



# Restauration d'image de contours incomplets par modélisation par champs de Markov

Sabine Urago, Marc Berthod, Josiane Zerubia

## ► To cite this version:

Sabine Urago, Marc Berthod, Josiane Zerubia. Restauration d'image de contours incomplets par modélisation par champs de Markov. [Rapport de recherche] RR-1688, INRIA. 1992. inria-00076923

**HAL Id: inria-00076923**

**<https://hal.inria.fr/inria-00076923>**

Submitted on 29 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# INRIA

UNITÉ DE RECHERCHE  
INRIA-SOPHIA ANTIPOLIS

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Rocquencourt  
B.P. 105  
78153 Le Chesnay Cedex  
France  
Tel. (1) 39 63 55 11

## Rapports de Recherche

1 9 9 2



25<sup>ème</sup>  
anniversaire

N° 1688

*Programme 4*  
*Robotique, Image et Vision*

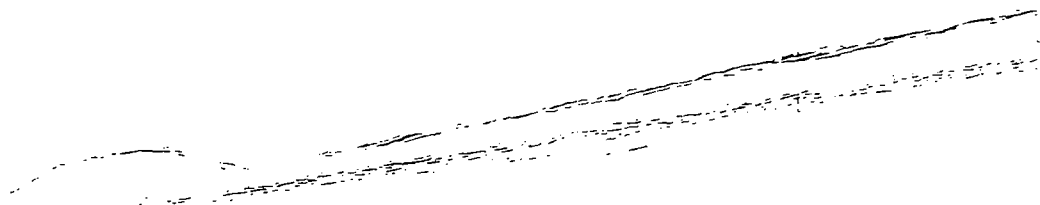
### RESTAURATION D'IMAGE DE CONTOURS INCOMPLETS PAR MODELISATION PAR CHAMPS DE MARKOV

Sabine URAGO  
Marc BERTHOD  
Josiane ZERUBIA

Mai 1992



★ R R . 1 6 8 8 ★



**RESTORATION OF INCOMPLETE CONTOUR IMAGES  
USING MARKOV RANDOM FIELDS.**

**RESTAURATION D'IMAGE DE CONTOURS  
INCOMPLETS PAR MODELISATION  
PAR CHAMPS DE MARKOV.**

**Sabine URAGO, Marc BERTHOD et Josiane ZERUBIA**

**INRIA, Sophia-Antipolis  
2004 Route des Lucioles  
BP 93  
06902 Sophia Antipolis Cedex  
France**

**Tel : (33) 93 65 78 57  
Fax : (33) 93 65 76 43**

## **Abstract :**

In this paper, we describe an algorithm which allows the restoration of images with incomplete contours. We use Markov random fields and Gibbs distributions.

In order to restore the contours, we define a few criterions which have to be optimized. Deterministic (Iterated Conditional Mode) or stochastic relaxation (Gibbs sampler) algorithms generate a configuration in which the contours are completed.

Our study shows how to modify and extend the method proposed by J.L Marroquin in 1989 to enable the processing of real and noisy images.

To illustrate this algorithm, several examples are given including synthetic, noisy and real (indoors and satellite : SPOT) images.

## **Résumé :**

Dans ce rapport, nous présentons un algorithme permettant de restaurer des images de contours incomplets. Nous utilisons une modélisation markovienne et des distributions de Gibbs.

Pour ce faire, certains critères à optimiser sont déterminés afin de compléter les parties manquantes.

La mise en oeuvre de méthodes déterministes ("Iterated Conditional Mode") ou stochastiques (échantillonneur de Gibbs) de relaxation permet d'aboutir à une configuration stable dans laquelle les contours sont complétés.

Notre travail consiste à modifier un algorithme proposé par J.L Marroquin en 1989 et à l'étendre dans le but de traiter des images bruitées et de scènes réelles.

Pour illustrer cette méthode, nous présentons des résultats sur plusieurs types d'images : des images synthétiques bruitées et des images réelles (images d'intérieur, images satellite : SPOT).

1	INTRODUCTION.....	5
2.	RESTAURATION D'IMAGE.....	6
2.1.	Description des modèles pour l'image et la dégradation.....	6
2.2.	Champs de Markov, distribution de Gibbs et Théorème de Bayes.....	7
2.3.	Méthodes d'optimisation.....	10
2.3.1.	Description des méthodes stochastiques de relaxation basées sur les distributions de Gibbs.....	11
2.3.1.1	L'algorithme de Metropolis.....	11
2.3.1.2	L'échantillonneur de Gibbs.....	13
2.3.2	Description de deux méthodes déterministes.....	14
2.3.2.1	Le "GNC" (Graduated Non-Convexity).....	14
2.3.2.2	"ICM" ("Iterated Conditional Mode").....	16
3.	CHAMPS MARKOVIENS D'ELEMENTS DE CONTOURS.....	17
3.1.	choix d'un vecteur de champ de Markov.....	18
3.1.1.	Première composante : les états.....	18
3.1.2	Deuxième composante : la direction macroscopique.....	24
3.1.2.1	Définition.....	24
3.1.2.2	Subdivision en intervalles homogènes.....	24
3.1.2.3	Exemple :.....	25
3.2	Fonction d'énergie.....	27
3.2.1	Notations.....	28
3.2.2	Fonctions potentielles.....	28
3.2.2.1	Fonction V1.....	29
3.2.2.2	Fonction V2.....	30
3.2.3	Energie totale.....	31
3.2.4	Rôle des paramètres dans la fonction d'énergie.....	31
3.3	Implantation pratique.....	33
3.4	Application pour la restauration d'image de contours.....	35
3.4.1	Initialisation des états.....	35
3.4.2	Initialisation des directions macroscopiques.....	35
3.4.3	Exemple simple de prolongements et de raccourcissements de lignes.....	39
3.4.3.1	Implantation parallèle.....	39
3.4.3.2	Mise en oeuvre de l'algorithme sur une image de taille réduite.....	40
3.4.4	Période de phases de prolongements et de raccourcissements des lignes.....	53
3.4.4.1	Période stable.....	53
3.4.4.2	Période variable.....	53
3.4.4.3	Application des deux méthodes sur un exemple simple.....	54
3.4.4.4	Problèmes rencontrés.....	54
3.4.5	Comparaison de deux algorithmes : Echantillonneur de Gibbs et ICM.....	58
3.4.6	Modifications effectuées dans le but de traiter des images bruitées.....	60

3.5	Résultats de simulations. ....	64
3.5.1	Mise en oeuvre sur la machine à connexions : CM2. ....	64
3.5.2	Résultats sur des images synthétiques non bruitées. ....	64
3.5.3	Résultats sur des images synthétiques bruitées. ....	69
3.5.4	Résultats sur des images de scènes réelles. ....	71
3.5.5	Temps d'exécution.....	79
4.	Conclusion.....	80
5.	Références.....	81

## 1 INTRODUCTION.

Dans ce rapport, nous présentons un algorithme permettant la restauration d'image de contours incomplets (les méthodes de détections de contours classiques ne permettant pas toujours des extractions de contours complets) cf [9], [10], [7], [2]. Le modèle utilisé est celui des champs de Markov ainsi que les distributions de Gibbs qui leur sont associées, afin d'aboutir à une configuration stable dans laquelle les contours sont complétés. L'algorithme implanté met en oeuvre soit un processus stochastique, basé sur l'échantillonneur de Gibbs [4], soit une méthode déterministe : "ICM" (Iterated Conditional Mode) [1].

Nous avons modifié et étendu un algorithme initialement proposé par J.L. Marroquin en 1989 [5] [6], afin de pouvoir traiter des images réelles (en particulier des images SPOT) pouvant être parfois bruitées.



## **2. RESTAURATION D'IMAGE.**

Restaurer une image signifie reconstituer, à partir d'une image dégradée, l'image originale ou de bonne qualité.

La dégradation d'une image peut avoir plusieurs origines. En effet, elle peut être provoquée soit par le capteur, soit par un bougé de l'appareil photographique, une défocalisation ou une diffraction. L'image détériorée est en général, floue ou bruitée.

Pour résoudre des problèmes tels que la restauration ou la segmentation d'images, des méthodes bayésiennes peuvent être mises en oeuvre. Ces techniques sont utilisables même lorsque la dégradation est non linéaire, mais nécessitent beaucoup de calculs.

**Pour restaurer une image, différentes étapes sont en fait nécessaires :**

Il faut tout d'abord, **choisir un modèle pour l'image et la dégradation**, puis **définir un critère à optimiser** pour restaurer l'image.

Pour cela, on utilise une certaine connaissance sur la nature de la solution. D'après le théorème de Bayes, cet a priori est utilisé pour décrire une forme de distribution de probabilité. La solution du problème est alors donnée par **le maximum de la densité de probabilité de l'image originale connaissant l'image dégradée.**

Il faut ensuite choisir une méthode permettant de trouver ce maximum.

### **2.1. Description des modèles pour l'image et la dégradation.**

Notons **X** l'image originale et **G** l'image dégradée.

Supposons que **X** et **G** soient deux images carrées.

Quelques notations :

$(m \times m) = M$  la taille des images.

$IND_m$  l'ensemble des pixels de l'image :

$IND_m = \{ (i,j) \text{ tels que } 1 \leq i,j \leq m \}$

• L'image originale **X** peut être modélisée de la manière suivante :

$$X = (F,L)$$

où

**F** est la matrice des niveaux de gris observés.

**L** représente la matrice duale des éléments de contours : la matrice des processus de ligne.

Notations :

$F = \{ F_{i,j} \}$   $(i,j)$  appartenant à  $IND_m$   
avec  $F_{i,j}$  est le niveau de gris du pixel  $(i,j)$ .

$W = (f,l)$  pour  $g$  fixé.

{  $F=f$  } signifie {  $F_{i,j} = f_{i,j}$  pour tout  $(i,j)$  appartenant à  $IND_m$  }

- L'image dégradée  $G$  peut se représenter dans le cas général par :

$$G = \Omega(H(F)) \ominus B$$

où

$H$  est la matrice correspondant au flou.

$\Omega$  est une transformation non linéaire.

$B$  correspond au bruit.

$\ominus$  est soit une addition, soit une multiplication.

## 2.2. Champs de Markov, distribution de Gibbs et Théorème de Bayes.

Pour restaurer l'image  $G$ , il faut donc définir un critère à optimiser. Pour ce faire un apprentissage est nécessaire et permet de fournir des informations a priori servant à définir une forme de distribution de probabilité.

Une classe de modèles probabilistes, qui est très commode pour la construction des densités de probabilités précédentes, est la classe des champs de Markov.

En effet, certaines propriétés des **champs de Markov** sont intéressantes :

- = Les caractéristiques, à un site donné, dépendent uniquement des interactions locales.

La probabilité d'avoir  $f_{ij}$  (un niveau de gris  $f$  pour le site  $(i,j)$ ) connaissant tous les niveaux de gris de l'image est égale à la probabilité d'avoir  $f_{i,j}$  connaissant toutes les intensités voisines.

$$P(F_{i,j}=f_{i,j} / F_{k,l}=f_{k,l} \text{ avec } (k,l) \neq (i,j)) = P(F_{i,j}=f_{i,j} / F_{k,l}=f_{k,l} \text{ avec } (k,l) \in \theta_{i,j})$$

$\theta_{i,j}$  représente le **voisinage du pixel  $(i,j)$** , il peut être défini en utilisant la quatre connexité (champ de Markov du premier ordre), la huit connexité (champ de Markov du deuxième ordre) etc...

**Exemples :**

- En utilisant des **champs de Markov du premier ordre** (quatre connexité), le voisinage du pixel  $(i,j)$ ,  $\theta_{i,j}$  est défini par :

$$\theta_{i,j} = \{ (i,j-1), (i,j+1), (i-1,j), (i+1,j) \}$$

alors qu'avec des **champs de Markov du deuxième ordre** (huit connexité), on a :

$$\theta_{i,j} = \{(i-1,j-1), (i-1,j), (i-1,j+1), (i,j-1), (i,j), (i,j+1), (i+1,j-1), (i+1,j), (i+1,j+1)\}$$

Champs de Markov  
du premier ordre :

i\j	j-1	j	j+1
i-1		•	
i	•	X	•
i+1		•	

Champs de Markov  
du deuxième ordre :

i\j	j-1	j	j+1
i-1	•	•	•
i	•	X	•
i+1	•	•	•

X représente le pixel (i,j)  
• sont les voisins du pixel (i,j)

Plus généralement, le choix de la **c-connexité** définit un système de voisinage  $\theta_{i,j}$  de la forme :

$$\theta_{i,j} = \{ \text{voisi}_{i,j}, \text{ avec } (i,j) \text{ appartenant à } \text{IND}_m \}$$

avec

$$\text{voisi}_{i,j} = \{ (k,l) \text{ appartenant à } \text{IND}_m \text{ et tels que}$$

$$0 < (k-i)^2 + (l-j)^2 \leq c \}$$

= D'après le **théorème de Hammersley-Clifford** : un champ est de Markov si et seulement si sa distribution est une **distribution de Gibbs** de la forme :

$$P(X=W) = \frac{1}{Z} \exp\left(\frac{-U(W)}{T}\right)$$

et

$$P(X=W) > 0$$

avec

$$U(W) = \sum_{c \in C} V_c(W)$$

où :

**U(W)** est une fonction d'énergie, c'est la somme des potentiels sur l'ensemble des cliques (voir définition).

$c$  est une clique, et  $C$  est l'ensemble de toutes les cliques.

$V_c$  est le potentiel de clique.

$Z$  est une constante de normalisation, appelée fonction de partition :

$$Z = \sum_{\text{toutes les configurations } W} \exp\left(\frac{-U(W)}{T}\right)$$

**Définition :**

Une clique est soit un seul site, soit un ensemble de sites deux à deux voisins.

Le but est alors de maximiser la densité de probabilité de l'image originale connaissant l'image dégradée.

D'après le **théorème de Bayes**, cette densité de probabilité  $P(X=W / G=g)$  s'écrit de la manière suivante :

$$P(X=W / G=g) = \frac{P(G=g / X=W) P(X=W)}{P(G=g)}$$

- $P(G=g)$  est un terme constant car  $g$  est fixé.
- $P(X=W)$  est la probabilité a priori d'occurrence de  $W$  et s'exprime de la manière suivante :

$$P(X=W) = \frac{1}{Z} \exp\left(\frac{-U(W)}{T}\right)$$

et

$$U(W) = \sum_{c \in C} V_c(W)$$

- $P(G=g / X=W)$  est la densité de probabilité de l'image dégradée connaissant l'image originale.  
Dans le cas d'une modélisation gaussienne, on a :

$$P(G=g / X=W) = (1 / \Delta \sqrt{2\pi}) \exp(-||g - \mu||^2 / 2\Delta^2)$$

où  $\mu$  est la moyenne et  $\Delta$  l'écart-type.

- La densité de probabilité de l'image originale connaissant l'image dégradée peut donc s'écrire :

$$P(X=W / G=g) = \frac{1}{Z'} \exp \left( \frac{-U'(W)}{T} \right)$$

avec

$$U'(W) = \frac{U(W)}{T} + ||g - \mu||^2 / 2\Delta^2$$

et

$$Z' = Z \Delta \sqrt{2\pi}$$

Maximiser cette densité de probabilité (critère du maximum a posteriori : MAP) est équivalent à minimiser l'énergie.

### 2.3. Méthodes d'optimisation.

Le problème est de maximiser la densité de probabilité  $P$  (ou minimiser une énergie  $U$  non convexe) qui peut posséder plusieurs maxima (ou minima) locaux. On souhaite en général, trouver l'extremum global.

Plusieurs solutions sont alors envisageables :

Des **méthodes stochastiques** [4], [2], [7] génèrent des séquences d'images qui convergent asymptotiquement vers le minimum global. Ces algorithmes autorisent aux hautes températures des sauts d'énergie qui permettent de ne pas être piégé dans des minima locaux. La mise en œuvre de ces algorithmes nécessite un temps de calcul élevé, mais a l'avantage d'atteindre un maximum global lorsque la température tend vers zéro.

Voici quelques exemples de méthodes stochastiques :

- **L'algorithme de Metropolis** [7].
- **L'échantillonneur de Gibbs** [4], [7].

Contrairement à ces algorithmes, des **méthodes déterministes** créent des séquences d'images qui convergent vers un minimum local. Elles sont sous-optimales mais relativement rapides.

Voici quelques exemples de méthodes déterministes :

- **GNC ("Graduated Non-Convexity")** [2].
- **MFA ("Mean Field Annealing")** [3].
- **ICM ("Iterated Conditional Mode")** [1].

### 2.3.1. Description des méthodes stochastiques de relaxation basées sur les distributions de Gibbs.

Les deux algorithmes stochastiques de relaxation : l'algorithme de Metropolis [7] et l'échantillonneur de Gibbs [4] peuvent être décrits de la manière suivante :

#### 2.3.1.1 L'algorithme de Metropolis.

Imaginons un processeur placé en chaque site  $s$  de l'image, chaque processeur étant connecté aux processeurs des sites voisins. L'état de cette "machine", évolue par des changements discrets au cours du temps, créant ainsi des séquences d'images. Pour obtenir cette évolution, le temps doit être discrétisé ( $t=0, t=1\dots$ ).

A l'instant initial une **température** assez élevée est fixée, et va décroître progressivement au cours du temps, jusqu'à atteindre une valeur nulle.

A une température fixée, tous les sites sont étudiés séquentiellement en utilisant l'algorithme suivant :

A l'instant  $t$ , l'état de chaque processeur placé au site  $s$  correspond à une **variable aléatoire notée  $X_s(t)$** . Cette variable dépend du modèle choisi pour l'image originale, elle peut par exemple correspondre aux niveaux de gris et aux processus de lignes du site  $s$ .

Soit  $M$ , le nombre de sites de l'image, la **configuration totale au temps  $t$**  peut donc s'écrire :

$$X(t) = (X_1(t), X_2(t), \dots, X_N(t))$$

La configuration initiale, notée  $X(0)$ , est totalement arbitraire.

A un instant  $t$  donné, la valeur d'un seul site peut être modifiée. Les configurations  $X(t-1)$  et  $X(t)$  diffèrent donc au plus d'une coordonnée. Notons  $n_1, n_2, \dots$  la séquence dans laquelle les sites sont étudiés : **Au temps  $t$ , le seul changement possible aura lieu au site  $n_t$ .**

On choisit aléatoirement un état  $X$  (appartenant à un ensemble de variables aléatoires déterminé) pour le site  $n$ .

La nouvelle configuration possible est donc  $X'(t)$  définie par :

$$\begin{aligned} X'_i(t) &= X_i(t-1) \text{ pour } i \neq n_t \\ \text{et} \\ X'_{n_t}(t) &= X \end{aligned}$$

On calcule ensuite la différence d'énergie :

$$\Delta U = U' - U$$

où  $U$  (respectivement  $U'$ ) représente l'énergie avant (respectivement après) l'éventuel changement.

Deux solutions sont alors possibles :

- Si  $\Delta U \leq 0$  ( si le nouvel état permet de diminuer l'énergie) ou si (  $\Delta U > 0$  et  $\exp(-\frac{\Delta U}{T}) \geq \alpha$  ) avec  $\alpha$  variable aléatoire de distribution uniforme, alors le changement est effectué.

La nouvelle configuration pour l'instant  $t$  est donc  $X(t)$  avec :

$$\begin{aligned} X_i(t) &= X_i(t-1) \text{ pour } i \neq n_t \\ &\text{et} \\ X_{n_t}(t) &= X \end{aligned}$$

- Sinon la valeur du pixel reste inchangée :  $X(t) = X(t-1)$ .

Un choix, (par exemple du niveau de gris et des éléments de contours), est ainsi effectué, pour le site courant, en tolérant de réduire le maximum de la densité de probabilité précédemment obtenue.

#### Remarques :

Les distributions conditionnelles locales dépendent d'un paramètre global la température. En effet, elles sont de la forme :

$$P = \frac{1}{Z} \exp(-\frac{U}{T})$$

A haute température, la distribution est pratiquement uniforme, ce qui entraîne un aspect chaotique de l'image mais permet d'éviter la convergence vers un maximum local. Par contre, une température basse permet d'isoler les états correspondants à un bas niveau d'énergie. L'image obtenue a alors un aspect beaucoup plus régulier.

En pratique, la décroissance de la température doit être suffisamment lente, ce qui rend l'algorithme très long.

La température à la  $k$ -ième itération  $T(k)$  doit être proportionnelle à  $\frac{1}{\log(k)}$ .

Un nombre d'itérations ou un critère d'arrêt est donc fixé.

Notons aussi que la configuration initiale  $X(0)$  n'intervient pas pour la limite obtenue lors de la convergence de l'algorithme.

La propriété de Markov permet de mettre en oeuvre cet algorithme de façon parallèle. En effet, le traitement est le même en chaque site et dépend uniquement des sites voisins. Il suffit donc d'étudier parallèlement des sites "suffisamment éloignés" (non voisins). Le temps nécessaire pour les mises à jour est donc peu élevé sur une machine parallèle.

### 2.3.1.2 L'échantillonneur de Gibbs.

Le principe de l'algorithme de l'échantillonneur de Gibbs [4] est d'échantillonner suivant une loi de probabilité  $P$ . Ici cette probabilité est la probabilité  $P(G/X)$  (cf § 4.2) que nous devons maximiser.

**Voici une description de cet algorithme :**

La configuration initiale  $X(0)$  est choisie aléatoirement.

On fixe également une température initiale  $T_0$ . Au cours du temps cette température diminue lentement jusqu'à atteindre la valeur nulle.

A une température donnée, tous les sites sont étudiés séquentiellement. Les deux configurations  $X(t)$  et  $X(t-1)$  diffèrent donc au plus d'une unique composante  $X_s$  (correspondante au site  $s$ ) mise à jour de la manière suivante :

On calcule les énergies locales pour chacun des  $N$  états ( $e_i$ ,  $i=1..N$ ) possibles ( $N$  variables préalablement déterminées).

Puis on détermine les  $N$  probabilités correspondantes :

$$P(e_i) = \exp\left(\frac{-U(e_i)}{T}\right)$$

Ces  $N$  probabilités sont normalisées :

$$P\_norma(e_i) = \frac{P(e_i)}{\sum_{i=1}^N P(e_i)}$$

Le choix de la nouvelle variable aléatoire pour le site  $s$  est effectué par un tirage aléatoire :

$$\text{Si } \left(\sum_{i=1}^{k-1} P\_norma(e_i)\right) < \alpha \leq \sum_{i=1}^k P\_norma(e_i)$$

où  $\alpha$  est une variable aléatoire de distribution uniforme.

Alors la nouvelle configuration pour l'instant  $t$  est :

$$\begin{aligned} X_i(t) &= X_i(t-1) \text{ pour } i \neq s \\ \text{et} \\ X_s(t) &= e_k \end{aligned}$$

Les remarques concernant cet algorithme sont identiques à celles effectuées pour l'algorithme de Metropolis.



## 2.3.2 Description de deux méthodes déterministes.

### 2.3.2.1 Le "GNC" (Graduated Non-Convexity).

Le but est le même que précédemment : minimiser l'énergie  $E$  non-convexe.

Le principe de cette méthode est de transformer le problème initial en un autre problème que l'on sait résoudre.

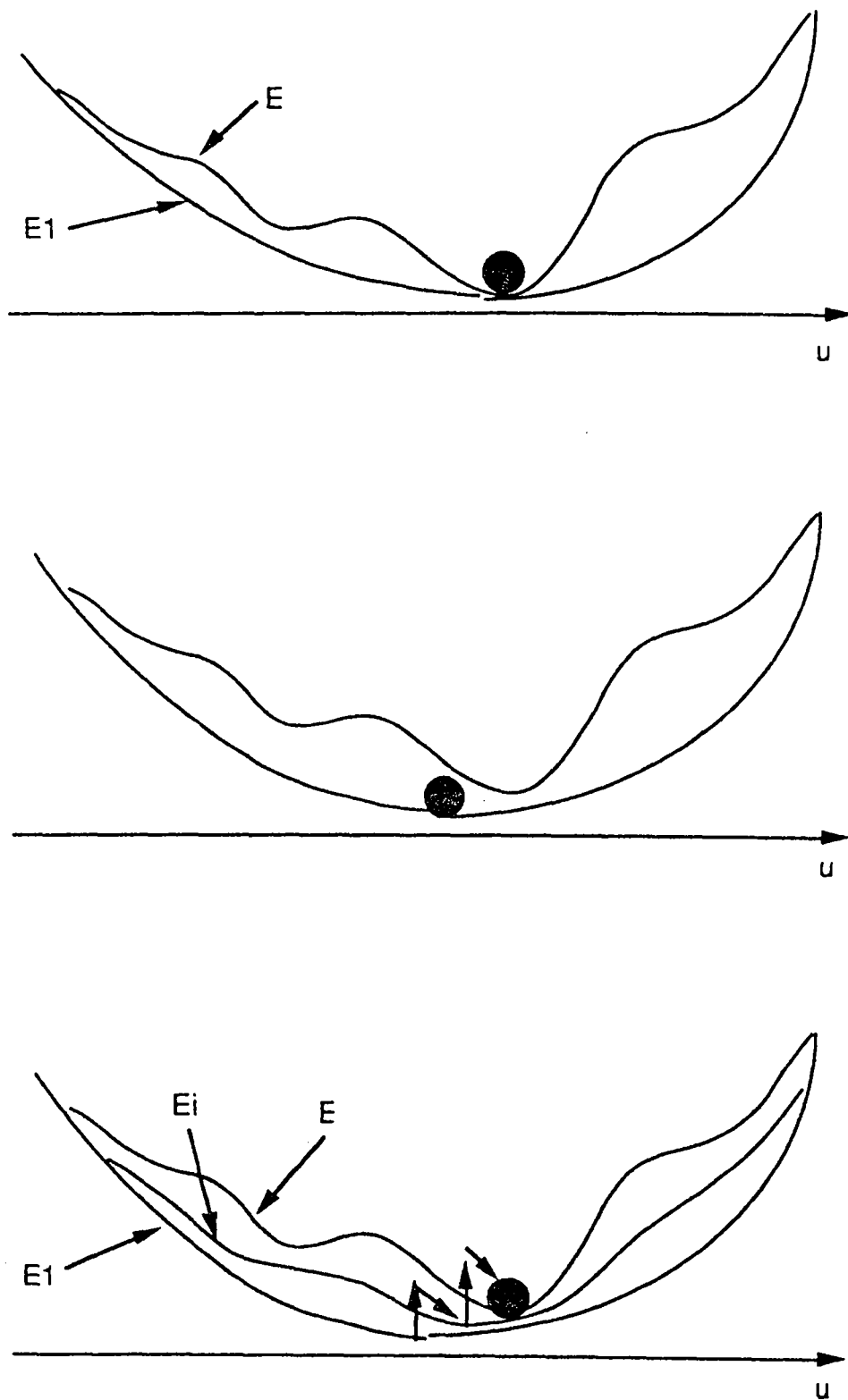
L'algorithme est basé sur la recherche d'une fonction  $E_1$ , qui est une approximation convexe de la fonction d'énergie  $E$ .

On exhibe une famille de fonctions  $(E_i)$ , où  $i$  décroît de 1 à 0. La fonction  $E_0$  est équivalente à  $E$ , et la seule fonction de la famille  $(E_i)$  convexe est la fonction  $E_1$ .

Pour plus de détails, on peut se référer au livre de Blake & Zisserman [2] et [8].

Voici une schématisation de la méthode du GNC :

Figure 0 : Algorithme du GNC ("Graduated Non-Convexity")  
d'après [2]



### 2.3.2.2 "ICM" ("Iterated Conditional Mode").

Cet algorithme [1] est un simulateur de Gibbs à température nulle :

On choisit la configuration initiale de telle sorte qu'elle soit la plus proche d'une configuration optimale.

Tous les sites sont également étudiés de manière séquentielle.

Pour mettre à jour le site  $s$ , on calcule, comme dans l'algorithme de l'échantillonneur de Gibbs, les énergies locales pour chacun des  $N$  états ( $e_i$ ,  $i=1..N$ ) possibles ainsi que les  $N$  probabilités correspondantes.

**L'état choisi pour le site  $s$  est alors celui qui correspond à la probabilité maximale.**

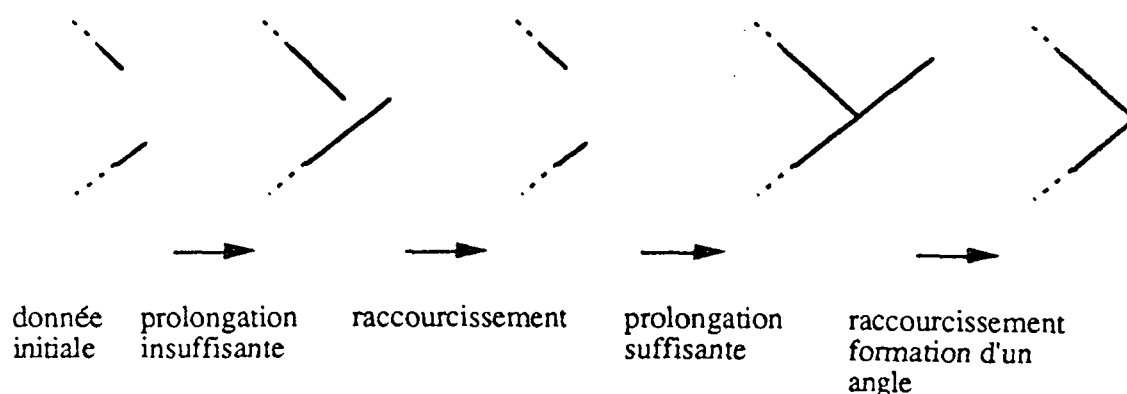
Cet algorithme peut, de même que les algorithmes décrits précédemment, être mis en oeuvre de façon parallèle.

Il est déterministe car il ne fait intervenir aucun tirage aléatoire. Son critère d'arrêt peut être déterminé simplement. En effet, on ne réitère les étapes décrites que si le nombre d'états modifiés est supérieur à un certain seuil (suffisamment petit).

### 3. CHAMPS MARKOVIENS D'ELEMENTS DE CONTOURS.

Le but est de reconstituer la totalité des frontières des objets d'une image à partir de données incomplètes. Pour ce faire [5],[6], on effectue successivement des prolongations et des raccourcissements des lignes de contours données initialement, dans le but d'obtenir une figure stable, où des "angles" (connexions entre deux lignes de directions macroscopiques différentes) sont déterminés et empêchent les variations de la longueur des lignes. Ainsi en présence de deux "morceaux" de contours non parallèles, ces différentes étapes peuvent être schématisées de la façon suivante :

Figure 1 : Exemple de reconstitution de contours



Nous pouvons constater dans ce schéma que l'amplitude des étirements et des raccourcissements varient au cours du temps. En fait plusieurs méthodes peuvent être mises en oeuvre, mais des explications plus détaillées seront données par la suite. L'image restaurée apparaît lorsque tous les angles sont déterminés. Ces étapes consécutives imposent de restreindre l'ensemble des images à compléter. En effet, les images doivent contenir uniquement des données manquantes, ce qui signifie que pour les images bruitées, un traitement préliminaire doit être effectué.

Pour restaurer l'image de contours, il est nécessaire de définir les représentations et les différents algorithmes appliqués. Pour cela, on détermine tout d'abord une classification des éléments constituant les frontières, ainsi qu'une caractérisation de l'image restaurée.

Pour la première fois en 1984, Geman et Geman [4] ont décrit par des champs de Markov, des niveaux de gris ainsi que des éléments duaux utilisant la connaissance sur les contours. Ces éléments lignes, sont en fait définis comme des segments ayant quatre orientations possibles. Geman et Geman ont également défini une fonction d'énergie associée et ont mis en oeuvre cette méthode pour segmenter une image.

Mais cette approche peut paraître insuffisante pour restaurer une image de contours incomplets. En effet, si l'image initiale ne contient que quelques éléments de contours

assez éloignés les uns des autres, cette méthode ne permet pas de combler les manques d'information importants, et donc de reconstituer correctement le contour. Ceci est dû au fait que seules quatre orientations sont possibles pour les éléments de contour ( $0^\circ, +45^\circ, +90^\circ, -45^\circ$ ).

En fait, il est nécessaire d'inclure pour chaque site, une information sur la direction macroscopique dans laquelle le contour est sensé être prolongé, ceci dans le but de combler correctement les trous créés par le manque d'information.

Ici, l'image à restaurer est modélisée par un champ de Markov. Pour chaque site, on mémorise au total, deux informations. Le champ de Markov est donc un vecteur qui a pour **première composante le type d'élément de contour**, alors que la **seconde** correspond à la **direction macroscopique de la ligne de contour**. L'état de chaque site ne dépend donc pas de tous les sites de l'image, mais uniquement des sites avoisinants. Une propriété intéressante est que ces sites voisins ne contiennent pas uniquement des informations locales, mais détiennent également une connaissance plus globale : la direction macroscopique.

Ces deux informations sont alors prises en compte par les potentiels de clique servant à calculer l'énergie. Une distribution de Gibbs est donc ainsi déterminée afin de mettre en oeuvre une méthode stochastique d'optimisation : l'**algorithme de l'échantillonneur de Gibbs** [4] est utilisé dans le but minimiser cette énergie. Une modification apportée à la méthode proposée par J.L Marroquin [5], [6] est la mise en oeuvre d'un algorithme déterministe : **ICM** [1]. Ces méthodes effectuent à un instant donné une mise à jour de la valeur d'un site connaissant les états des sites voisins. Cette nouvelle valeur est plus précisément déterminée par le calcul des probabilités conditionnelles des états possibles pour le site en question connaissant les états des sites voisins. Ceci permet d'obtenir une configuration particulière à l'état d'équilibre dans laquelle les contours sont reconstitués.

Commençons tout d'abord par décrire le champ de Markov utilisé.

### 3.1. choix d'un vecteur de champ de Markov.

Description des champs de Markov modélisant l'image restaurée [5], [6] :

#### 3.1.1. Première composante : les états.

La donnée initiale, l'image des contours incomplets est représentée par une matrice binaire (la valeur nulle pour un pixel donné, indique l'absence de contour en ce point, et inversement pour une valeur de un).

La classification des éléments de contours se compose de 21 classes, chacune d'elles correspondant à un état admis. Chaque état est déterminé à partir d'une fenêtre de taille  $3 \times 3$  de l'image initiale. Pour chaque site de l'image d'entrée, on génère un état. On construit donc ainsi une **matrice d'éléments de contours** de même dimension.

Prenons un exemple :

Soit une fenêtre de dimension 3\*3, centrée sur le pixel (i,j) de l'image à restaurer, ayant pour valeur :

i\j	j-1	j	j+1
i-1	0	1	0
i	0	1	0
i+1	0	1	0

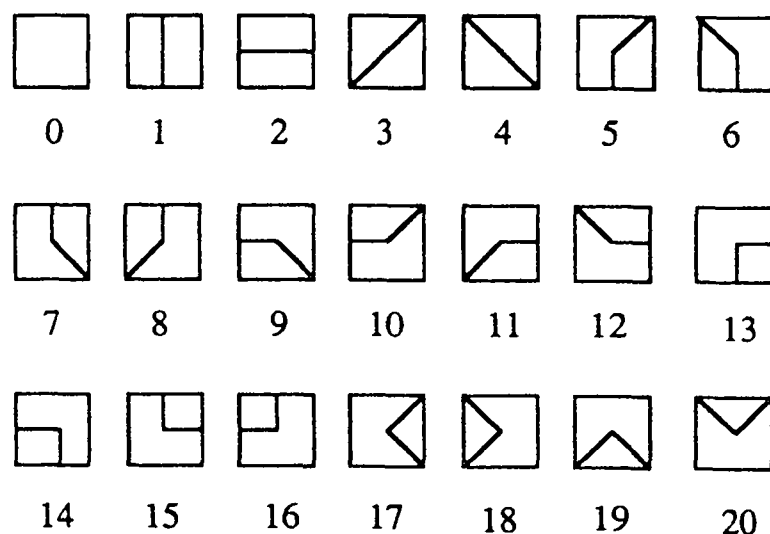
l'état du site (i,j) est alors défini comme étant l'état 1.

En numérotant les différents pixels de la fenêtre de dimension 3\*3 de la manière suivante:

i\j	j-1	j	j+1
i-1	0	1	2
i	3	4	5
i+1	6	7	8

on peut définir également l'état 1 en déterminant uniquement les pixels ayant pour valeur 1, c'est-à-dire par (1,4,7)

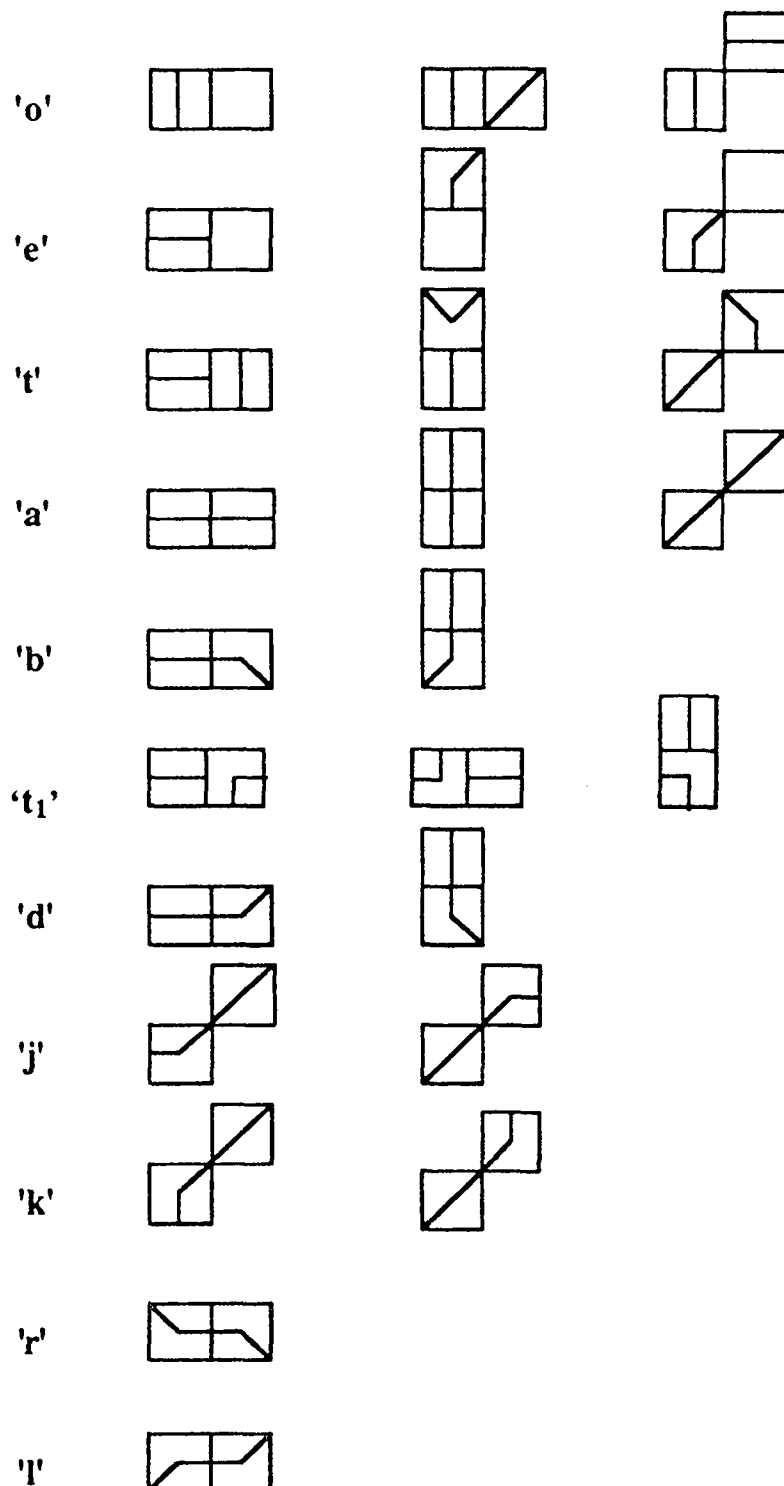
**Figure 2 : Description des 21 principaux états définis :**

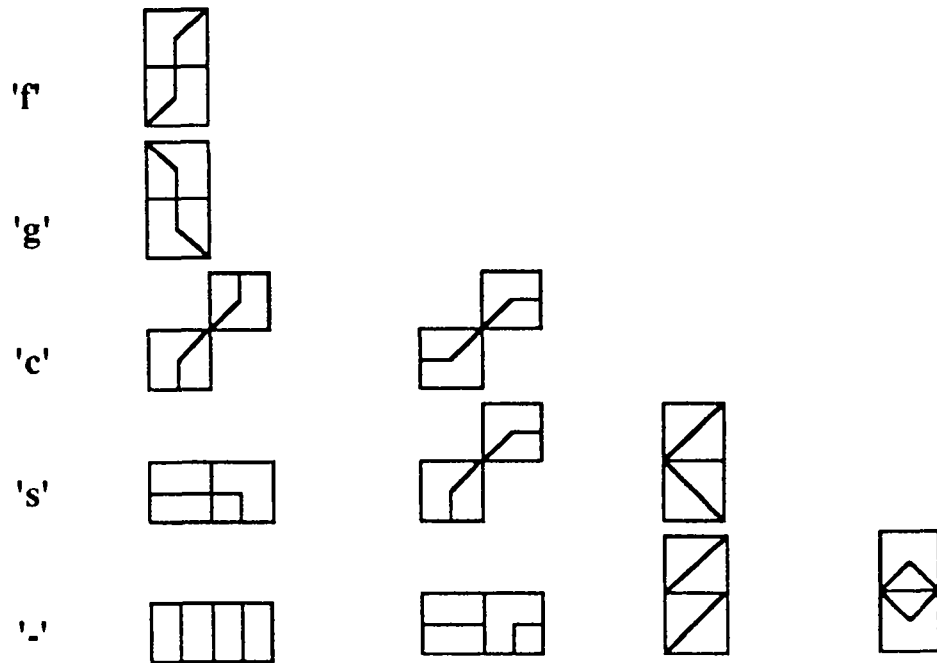


On établit également la convention suivante : Si la fenêtre (3\*3) de l'image initiale ne s'identifie à aucun de ces états prédéfinis, on suppose que l'état du site est nul.

Afin de prolonger les lignes de contours, il est indispensable de connaître les connexions entre deux états voisins. Pour cela, on détermine plusieurs classes correspondant à ces interactions locales.

**Figure 3 :** Exemples représentants chaque type de connexion :





Voici l'interprétation de chaque classe d'interaction :

Connexion indéterminée	: " o ".
Terminaison de lignes	: " e ".
Jonction en T (type t)	: " t ".
Jonction en T (type t <sub>1</sub> )	: " t <sub>1</sub> ".
Virage ayant un angle aigu	: " s ".
Continuation en ligne droite	: " a ".
Virage à droite	: " b ", " f ", " j ", " r ".
Virage à gauche	: " d ", " g ", " k ", " l ".
Virage gauche-droite	: " c ".
Connexion non admise	: " - ".

J.L Marroquin [5], [6] ne définit pas les connexions de type "t<sub>1</sub>", qui sont des T-jonctions, et il considère ces interactions comme étant "interdites", c'est-à-dire de type "-". Cette modification a été apporté dans le but d'obtenir, au bout d'un certain nombre d'itérations, une figure stable où tous les "angles" sont détectés.



Toutes les connexions possibles, obtenues à partir de tous les états voisins appartenant à des cliques horizontales ou diagonales, sont définies dans les deux tables suivantes.

Le premier tableau (table 1), représente toutes les interactions locales obtenues à partir d'un état "p" (indice de la ligne) et d'un état "q" (indice de la colonne), avec "p" et "q" appartenant à une clique horizontale.

Soient  $(X_p, Y_p)$  et  $(X_q, Y_q)$  les coordonnées respectives des sites ayant pour état "p" et "q", on a donc :

$$X_p = X_q - 1$$

et

$$Y_p = Y_q$$

Schématisation de la localisation des deux sites voisins :

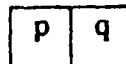


TABLE 1 : matrice des connexions pour des cliques horizontales.

p \ q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	o	o	e	o	o	o	o	o	o	e	e	o	o	o	e	o	e	o	o	o	o
1	o	-	t	o	o	-	-	-	-	t	t	o	o	-	-	-	-	-	-	-	-
2	e	t	a	t	t	t	t	t	t	b	d	t	t	t <sub>1</sub>	s	t <sub>1</sub>	s	t	-	-	-
3	o	o	t	-	s	-	s	-	-	t	t	-	s	-	-	-	-	-	-	-	-
4	o	o	t	s	-	-	-	-	s	t	t	s	-	-	-	-	-	-	-	-	-
5	o	-	t	o	s	-	s	-	-	t	t	o	s	-	-	-	-	-	-	-	-
6	o	-	t	o	o	-	-	-	-	t	t	o	o	-	-	-	-	-	-	-	-
7	o	-	t	s	o	-	-	-	s	t	t	s	o	-	-	-	-	-	-	-	-
8	o	-	t	o	o	-	-	-	-	t	t	o	o	-	-	-	-	-	-	-	-
9	o	o	t	s	o	o	o	o	s	t	t	s	o	-	-	-	-	-	-	-	-
10	o	o	t	o	s	o	s	o	o	t	t	o	s	-	-	-	-	-	-	-	-
11	e	t	d	t	t	t	t	t	t	s	l	t	t	-	s	-	s	t	-	-	-
12	e	t	b	t	t	t	t	t	t	r	s	t	t	-	s	-	s	t	-	-	-
13	e	-	s	-	-	-	-	-	-	s	s	-	-	-	-	-	-	-	-	-	-
14	o	-	t <sub>1</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	e	-	s	-	-	-	-	-	-	s	s	-	-	-	-	-	-	-	-	-	-
16	o	-	t <sub>1</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	o	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	o	-	t	-	-	-	-	-	-	t	t	-	-	-	-	-	-	-	-	-	-
19	o	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	o	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Le deuxième tableau (table 2), définit toutes les connexions possibles, en présence d'un état "p" (indice de la ligne) et d'un état "q" (indice de la colonne), appartenant à une clique diagonale.

on a alors :

$$X_p = X_q - 1$$

et

$$Y_p = Y_q + 1$$

Représentation des deux sites voisins :

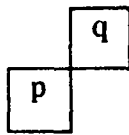


Table 2 : matrice des connexions pour des cliques diagonales.

p/q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	o	o	o	e	o	o	o	o	e	o	o	e	o	o	o	o	o	e	e	o	
1	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	t	t	o	
2	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	t	t	o	
3	e	t	t	a	t	t	t	t	k	t	t	j	t	t	t	t	t	s	s	t	
4	o	o	o	t	-	o	o	o	t	o	o	t	o	o	o	o	o	t	t	o	
5	e	t	t	k	t	t	t	t	c	t	t	s	t	t	t	t	t	s	s	t	
6	o	o	o	t	o	o	-	o	t	o	o	t	o	o	o	o	o	t	t	o	
7	o	o	o	t	o	o	o	-	t	o	o	t	o	o	o	o	o	t	t	o	
8	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	t	t	o	
9	o	o	o	t	o	o	o	o	t	-	o	t	o	o	o	o	o	t	t	o	
10	e	t	t	j	t	t	t	t	s	t	t	c	t	t	t	t	t	s	s	t	
11	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	t	t	o	
12	o	o	o	t	o	o	o	o	t	o	o	t	-	o	o	o	o	t	t	o	
13	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	t	t	o	
14	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	t	t	o	
15	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	t	t	o	
16	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	t	t	o	
17	e	t	t	s	t	t	t	t	s	t	t	s	t	t	t	t	t	s	s	t	
18	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	t	t	o	
19	o	o	o	t	o	o	o	o	t	o	o	t	o	o	o	o	o	t	t	o	
20	e	t	t	s	t	t	t	t	s	t	t	s	t	t	t	t	t	s	s	t	

Par symétrie, on obtient aisément les deux autres tables, correspondant aux cliques symétriques (verticales et diagonales).

Ces différentes connexions apparaissent dans la formule des potentiels de cliques, ainsi que les directions macroscopiques que nous allons déterminer.

### 3.1.2 Deuxième composante : la direction macroscopique.

#### 3.1.2.1 Définition.

Lors de la prolongation des lignes de contours, la connaissance de l'orientation de celles-ci est indispensable. En effet, on ne complète pas les contours par des éléments (états) choisis aléatoirement, mais par un état compatible avec la direction macroscopique de la ligne.

Cette direction est déterminée par un numéro compris entre (-1) et (11).

Pour un site donné, le **numéro de direction** peut être négatif dans plusieurs cas :

- état nul (absence de contour ou élément de contour indéfini).
- état supérieur à 12 (angle droit)
- **coin détecté** : un coin est localisé à l'intersection de deux lignes ayant des orientations différentes, ce qui implique la présence de plusieurs directions macroscopiques positives dans le voisinage d'un coin.

#### 3.1.2.2 Subdivision en intervalles homogènes.

Nous pouvons remarquer que tous les états ne sont pas compatibles avec toutes les connexions et toutes les directions macroscopiques données. De même, chaque connexion définie précédemment peut être associée uniquement avec certaines orientations.

Nous pouvons effectuer une **discrétisation des orientations**. En fait, chaque état possède une certaine pente pouvant être facilement associée à une direction macroscopique. Discrétiser les différentes orientations est équivalent à **discrétiser les pentes** correspondantes.

Pour ce faire, on subdivise donc l'ensemble de définition des pentes ( $]-\infty, +\infty[$ ), en sous-intervalles, chaque sous-intervalle étant associé à un numéro de direction (numéro variant de 1 à 11). Cette subdivision est créée dans le but suivant : **chaque numéro de direction correspondant à l'un de ces sous intervalles, doit être compatible avec un unique ensemble d'états et de connexions**. Un tel intervalle est dit homogène.

On définit également une valeur "**alpha**" représentant pour un site donné la **probabilité de tourner sur la probabilité de continuer tout droit**. C'est en fait, la probabilité d'avoir un état ayant un numéro compris entre 5 et 12, sur celle d'en avoir un compris entre 1 et 4.

Un exemple simple :

Prenons l'exemple d'une ligne droite composée uniquement d'éléments de contours numéro 3 (état 3 : /). Le fait que cette ligne de contour ne contienne aucun état supérieur à 4 (élément de contour formant un virage) implique qu'en un site situé sur cette droite, la probabilité de tourner est forcément nulle ainsi que la valeur "**alpha**".

Toutes ces informations sont regroupées dans la table suivante :

**TABLE 3 : Compatibilité des directions, des états et des connexions**

N°	Intervalles des pentes	Etats compatibles	Connexions ligne droite	Connexions virage	Connexions double virage
0	0	2	a		
1	$[0, 1/2]$	2,10,11	a	d,l	c
2	$[1/2, 1]$	3,10,11	a	j,c	l
3	1	3	a		
4	$[1, 2]$	3,5,8	a	k,c	f
5	$[2, \infty)$	1,5,8	a	b,f	c
6	$\infty$	1	a		
7	$(-\infty, -2]$	1,6,7	a	d,g	c
8	$[-2, -1]$	4,6,7	a	j,c	g
9	-1	4	a		
10	$[-1, -1/2]$	4,9,12	a	k,c	r
11	$[-1/2, 0]$	2,9,12	a	b,r	c

La deuxième colonne correspond aux différents intervalles définis à partir de la subdivision de l'ensemble des pentes. A chaque intervalle est associé un numéro de direction (colonne 1). A chacun de ces numéros correspond un unique ensemble d'états (colonne 2) et de connexions (colonnes 4,5 et 6). **Pour une ligne donnée, le numéro de direction, les états et les différentes interactions locales sont compatibles entre eux.**

Les connexions sont séparées en trois classes : La première (colonne 4) indique une continuation en ligne droite, la seconde (colonne 5) un tournant et la dernière (colonne 6) un double virage.

Voici un exemple détaillé, utilisant toutes les informations données par cette table :

### 3.1.2.3 Exemple :

**L'état 5** a une pente de valeur 2. Il est donc compatible avec les deux intervalles  $[1,2]$  et  $[2,\infty)$ , c'est-à-dire avec les numéros de direction 4 et 5.

**L'état 8** a les mêmes caractéristiques.

**L'état 3** possède une pente égale à 1 et une direction de  $\pi/4$ . Il est donc également associé au numéro de direction 4.

Ces trois états sont schématisés par :



Etat  
3



Etat  
5



Etat  
8

Les états 3, 5 et 8 correspondent au même numéro de direction, ils sont donc compatibles entre eux, et avec la direction 4.

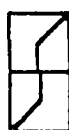
Quels types de connexion est-il possible d'obtenir à partir de ces trois états?

En fait, les seules interactions locales compatibles, pour cet exemple, sont les suivantes :

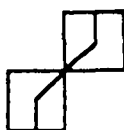
Continuation en ligne droite : " a "  
Virage : " k ", " f "  
Virage gauche-droite : " c "

En effet, nous pouvons les schématiser ainsi :

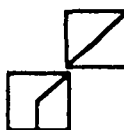
Connexions compatibles avec les états 3, 5 et 8.



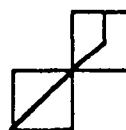
f



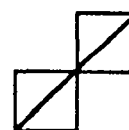
c



k



k



a

Prenons par exemple, la séquence d'états suivante (en gras), décrite avec les éléments 3, 5 et 8 :

							3
						3	
					5		
					8		
				3			
			3				
		3					
	5						
	8						
3							

Elle correspond à une **partie de l'image binaire de contour** (cf image donnée initialement) suivante :

							1
						1	
					1		
					1		
				1			
			1				
		1					
	1						
	1						
1							

### Remarques :

La ligne de contour, ayant comme numéro de direction 4, peut être constituée uniquement par des états 3,5 et 8.

Nous pouvons définir la valeur "alpha" qui correspond à la probabilité de tourner sur la probabilité de continuer tout droit. Pour cela, un dénombrement d'états inférieurs et supérieurs à 5 (éléments de contours droits et tournants) est nécessaire. dans cette séquence d'états.

Dans notre exemple, on trouve donc

$$\alpha = \frac{\text{probabilité de tourner}}{\text{probabilité de continuer tout droit}}$$

Ici, la probabilité de tourner (nombre de tournants sur le nombre total d'états considérés) est égale à 2/7, alors que celle d'aller droit est 3/7.

$$\alpha = \frac{2}{3}$$

## 3.2 Fonction d'énergie.

La fonction d'énergie est définie de telle sorte qu'elle minimise certains critères. Ces critères dépendent de l'image que nous souhaitons obtenir. Ici, le but étant de reconstituer une image de contours incomplets, l'énergie minimum doit être atteinte lorsque l'image des contours est entièrement reconstituée. Pour cela, les critères doivent être choisis de manière à **prolonger les lignes de contours, dans une direction macroscopique donnée, et de créer des angles (T-jonctions) afin d'obtenir des contours fermés.**

La fonction d'énergie est calculée à l'aide de deux fonctions potentielles V1 et V2.

Avant de définir ces deux fonctions, il est nécessaire de décrire les notations utilisées :

### 3.2.1 Notations.

**L** : ensemble des 21 états C représentés par la figure 2.

**D** : ensemble des numéros de directions macroscopiques D défini précédemment  
(  $D = [1,11] \cup \{-1\}$  )

$I_{ij} = I(q_i, q_j)$  correspond à la connexion formée par deux états voisins  $q_i$  et  $q_j$ .

Un **numéro de direction** donné, en un **site i** (noté  $d_i$ ) est compatible avec un unique ensemble composé au plus de quatre connexions. En effet, chaque numéro de direction est compatible au plus, avec une connexion représentant une continuation en ligne droite, deux virages et un virage gauche-droite (voir la table 3).

Ces trois classes de connexions sont notées :

- $SC_i = SC(d_i)$  Continuation en ligne droite.
- $T_i = T(d_i)$  Tournant.
- $CT_i = CT(d_i)$  Double virage.

De plus :

$C_i = C(d_i)$  représente l'ensemble des états compatibles avec la direction  $d_i$  (voir la table 3)

$\alpha_i = \alpha(d_i)$  est le quotient "alpha" (la probabilité de tourner sur celle de continuer tout droit).

On a :

$$\alpha_i = \frac{\text{probabilité de sélectionner un état } q \text{ appartenant à } T_i}{\text{probabilité de sélectionner un état } q \text{ appartenant à } SC_i}$$

### 3.2.2 Fonctions potentielles.

**V1** est une fonction réelle définie pour un site donné sur l'ensemble  $(L \times D)$ . En effet, elle dépend uniquement du vecteur  $(q_i, d_i)$  où  $q_i$  représente l'état du site et  $d_i$  sa direction macroscopique.

**V2** est une fonction potentielle de clique définie pour deux vecteurs  $(q_i, d_i)$  et  $(q_j, d_j)$  où  $i$  et  $j$  sont deux sites voisins. **V2** est donc une fonction réelle définie sur l'ensemble  $(L \times D \times L \times D)$ .

La fonction **V2** fait intervenir trois constantes réelles **VE**, **VT** et **VC** (cf § 5.2.4).

### 3.2.2.1 Fonction V1.

La fonction V1 est définie par :

$$\begin{aligned} V1(q_i, d_i) &= \begin{aligned} &\infty && \text{si } (d_i \geq 0 \text{ et } q_i \notin C_i) \\ &\ln(1+\alpha_i) && \text{si } (d_i \geq 0 \text{ et } q_i \in [1,4] \cap C_i) \\ &\ln\left(\frac{1+\alpha_i}{\alpha_i}\right) && \text{si } (d_i \geq 0 \text{ et } q_i \in [5,12] \cap C_i) \\ &0 && \text{sinon} \end{aligned} \end{aligned}$$

Quelques remarques sur le choix de cette fonction :

Cette fonction potentielle a une valeur infinie pour un site  $i$ , ayant un état  $q$  incompatible avec sa direction macroscopique  $d_i$ . Le choix de cette fonction potentielle favorise donc les états ayant une pente compatible avec la direction de la ligne à compléter.

Nous avons affecté, pour chaque élément de contours d'une même ligne, une valeur "alpha" correspondant à la probabilité de tourner sur la probabilité de continuer tout droit. Lorsque nous allons compléter les contours, nous allons propager cette information, car elle permet de prolonger les contours dans la bonne direction et avec des éléments adéquats.

En effet, la minimisation de cette fonction potentielle favorise la prolongation des contours le long d'une ligne de direction  $d_i$  par les éléments de contours "droits" (états compris entre 1 et 4) lorsque la ligne contient plus d'éléments droits que de tournants et inversement. En fait, lorsque le quotient " $\alpha_i$ " est inférieur à 1, c'est-à-dire lorsque la probabilité de tourner est inférieure à celle de continuer tout droit, on a :

$$\ln(1+\alpha_i) < \ln\left(\frac{1+\alpha_i}{\alpha_i}\right)$$

Or V1 a pour valeur  $\ln(1+\alpha_i)$  lorsque les éléments de contours sont "droits" (états compris entre 1 et 4), et  $\ln\left(\frac{1+\alpha_i}{\alpha_i}\right)$  pour les virages (états compris entre 5 et 12). On privilégie donc dans ce cas là, les états compris entre 1 et 4.

De plus, pour un site donné, la direction est négative lorsqu'il n'y a pas de contour (état 0) ou lorsque qu'un coin (plus précisément une "T-jonction") est détecté (état supérieur à 12 : formant un angle droit). Or la fonction V1 a comme valeur 0 pour les directions négatives. On favorise donc la formation des "T-jonctions".



### 3.2.2.2 Fonction V2.

La fonction potentielle V2 est définie pour une clique de taille 2 par :

$V2(q_i, d_i, q_j, d_j) =$	0	si ( $I_{i,j} = "o"$ )
	VE	si ( $I_{i,j} = "e"$ )
	VC	si ( $d_i = d_j$ ) et ( $q_i, q_j \in C_i$ ) et ( $I_{i,j} \in SC_i \cup T_i$ )
	$VC - \ln(\frac{1 + \alpha_i}{\alpha_i})$	si ( $d_i = d_j$ ) et ( $q_i, q_j \in C_i$ ) et ( $I_{i,j} = CT_i$ )
	VT	( si ( $I_{i,j} \notin \{ "o", "e", "-" \}$ et $d_k < 0$ et $q_k > 0$ pour au moins un $k \in \{ i, j \}$ ) ou ( si $I_{i,j} = "t"$ )
	VT <sub>1</sub>	si $I_{i,j} = "t_1"$
	$\infty$	sinon

Voici quelques caractéristiques de cette fonction :

- Lorsqu'une interaction est indéterminée ("o") le potentiel V2 est nul.
- Pour une connexion représentant la fin d'une ligne de contour de type "e" (connexion avec l'état "vide" 0), on affecte à V2 la valeur réelle VE.

Dans ces deux cas (interaction "o" ou "e"), nous pouvons remarquer que la direction macroscopique de la ligne n'intervient pas dans le calcul de V2.

- Lorsque la connexion de deux états de sites voisins ayant la même direction est une continuation en ligne droite ou un virage, V2 a pour valeur VC, constante réelle positive ou négative. Si cette interaction est une continuation en ligne droite, on a  $V2 + V1 = VC$ .
- De même, si les deux sites voisins possèdent la même direction et si la connexion est un virage gauche-droite alors  $V1 + V2 = VC$ .
- Pour les connexions dites "interdites" ("-"), V2 a une valeur infinie, ceci dans le but de rendre certaines configurations inaccessibles.
- VT est affecté pour toutes les connexions hormis les connexions de types "o", "e" et "-", lorsqu'au moins un des deux états possède une direction négative, c'est-à-dire en présence d'un angle.
- V2 a également la valeur VT lorsque la connexion entre les deux états est de type "t" ("T-jonction"), et il n'est pas nécessaire dans ce cas, que l'une des deux directions soit nulle.

- Pour une connexion de type "t<sub>1</sub>", V<sub>2</sub> a pour valeur VT<sub>1</sub>.

Ces deux derniers points n'apparaissent pas dans les travaux de J.L Marroquin [5] et [6], et favorisent l'obtention de contours fermés.

### 3.2.3 Energie totale.

On définit l'énergie totale "U" par la somme sur tous les sites de l'image, de la fonction potentielle V<sub>1</sub> et de la somme sur toutes les cliques de la fonction V<sub>2</sub> :

$$U = \sum_{i,j: ||i-j|| < 2} V_2(q_i, d_i, q_j, d_j) + \sum_i V_1(q_i, d_i)$$

i,j représentent des pixels voisins (appartenant à une même clique) de l'image.

D'après la distribution de Gibbs, la probabilité conditionnelle d'avoir pour le site i le vecteur (q<sub>i</sub>, d<sub>i</sub>) est proportionnelle à :

$$P = \exp \left( \frac{-1}{T} \left( \sum_{i,j: ||i-j|| < 2} V_2(q_i, d_i, q_j, d_j) + \sum_i V_1(q_i, d_i) \right) \right)$$

où T représente la température. Ici nous fixons ce paramètre à un, durant toutes les itérations.

### 3.2.4 Rôle des paramètres dans la fonction d'énergie.

Les trois constantes réelles VE, VT<sub>1</sub>, VT, VC utilisées dans la formule de l'énergie contrôlent les critères à optimiser afin de compléter une image de contours.

La constante VE a pour rôle d'empêcher la formation de nouvelle ligne et, dans la mesure du possible d'obtenir à l'équilibre une image de contours fermés.

On remarque que la constante VE doit forcément être positive. En effet, si dans le voisinage d'un site aucun contour n'est détecté, c'est-à-dire si tous les vecteurs des sites voisins sont de type : (0,-1), alors l'énergie obtenue avec le vecteur (q,d), où q un état non nul, est égale à "2\*VE" (deux connexions de terminaison), alors que celle obtenue avec le vecteur (0,-1) est égale à 0.

Si VE est négatif, on a plus de chance de sélectionner un état q non nul que l'état 0. On favorise donc la création de nouvelles lignes en un site où aucun contour n'a lieu d'être.

VT et VT<sub>1</sub> contrôlent la formation des angles et des "T-jonctions".

La connexion de type "a", c'est-à-dire la continuation en ligne droite, peut être observée uniquement en présence de deux mêmes états voisins compris entre 1 et 4 ("états droits") ayant la même direction. L'énergie pour ce site a alors, pour valeur VC.

De même, si les deux sites voisins ont une direction identique et si la connexion est un virage gauche-droite alors les fonctions potentielles sont :

$$V1 = \ln\left(\frac{1+\alpha_i}{\alpha_i}\right)$$

et

$$V2 = VC - \ln\left(\frac{1+\alpha_i}{\alpha_i}\right)$$

L'énergie pour le site a donc également pour valeur VC.

La valeur :  $\exp(-VC)$  intervient donc dans la probabilité conditionnelle de prolonger des lignes de contour.

La probabilité conditionnelle de tourner est donnée par :

$$\exp\left(\ln\left(\frac{\alpha_i}{1+\alpha_i}\right) - VC\right).$$

En effet, si

$$V1 = \ln\left(\frac{1+\alpha_i}{\alpha_i}\right) \quad \text{et} \quad V2 = VC$$

$$\text{alors l'énergie pour ce site est : } U = \ln\left(\frac{1+\alpha_i}{\alpha_i}\right) + VC.$$

Et la distribution de Gibbs :

$$P = \exp\left(\frac{-U}{T}\right) = \exp\left(\frac{\left(\ln\left(\frac{\alpha_i}{1+\alpha_i}\right) - VC\right)}{T}\right)$$

Pour la reconstruction de contours, les paramètres VE, VT et VC doivent prendre des valeurs particulières :

- VE doit être "très grand" et positif pour assurer que les contours soient fermés et éviter la génération de nouvelles lignes.
- VC, VT et VT<sub>1</sub> doivent être fixés de façon à privilégier la continuation en ligne droite par rapport à la formation des "T-jonctions". Ceci s'obtient en choisissant  $|VC| < VT$ .
- Le signe de VC détermine la prolongation ou le raccourcissement des lignes.

VC < 0 : rallongement des lignes.

VC > 0 : raccourcissement des lignes.

Dans la pratique, pour la reconstruction des contours, nous avons choisi les paramètres de telle sorte que :

$$|VC| < VT < VT_1 \ll VE$$

### 3.3 Implantation pratique.

Cette distribution de Gibbs ainsi déterminée permet de mettre en oeuvre une méthode stochastique d'optimisation : l'échantillonneur de Gibbs [4], ou une méthode déterministe : "ICM" [1] dans le but minimiser cette énergie. Ces méthodes effectuent à un instant donné une mise à jour de la valeur d'un site connaissant les états des sites voisins. Ce nouveau vecteur  $(q_i, d_i)$  est déterminé par le calcul des probabilités conditionnelles des différents vecteurs  $(q, d)$  (où  $q$  représente l'état et  $d$  la direction) pour le site connaissant les états des sites voisins et par l'algorithme sélectionné.

**Voici une mise en oeuvre possible :**

On effectue une mise à jour du vecteur  $(q_i, d_i)$  (état et direction pour le site  $i$ ) de manière séquentielle, pour chaque site " $i$ ". Pour cela, on **détermine les probabilités conditionnelles**, connaissant les vecteurs  $(q_j, d_j)$  des sites voisins, **pour tous les vecteurs  $(q_i, d_i)$  possibles** : c'est-à-dire pour les 21 états définis avec les 13 numéros de directions (les 12 numéros de directions déterminés dans la table 3 et la direction -1). On sélectionne ensuite, en appliquant par exemple l'algorithme de l'échantillonneur de Gibbs [4], le vecteur  $(q, d)$  qui minimise a priori la fonction d'énergie. On réitère cette méthode jusqu'à l'obtention d'une image stable.

Mais cette implantation est très coûteuse en temps de calcul. En fait, il est déjà plus intéressant de tester non pas tous les états avec toutes les directions, mais **uniquement tous les états avec les directions compatibles**. On remarque que l'unique direction compatible avec l'état 0 et les états supérieurs à 12 (états représentant des angles droits) est la direction -1.

**Un algorithme plus rapide peut être mis en oeuvre :**

D'après les propriétés des deux fonctions potentielles, uniquement certaines paires  $(q_i, d_i)$  (correspondant à une énergie finie) doivent être considérées pour la nouvelle mise à jour. En effet, les vecteurs associés à des probabilités conditionnelles nulles, ne sont jamais sélectionnés par l'algorithme de l'échantillonneur de Gibbs [4]. Il en est de même, si c'est l'algorithme "ICM" [1] car le vecteur sélectionné est celui qui correspond à la probabilité maximale.

En fait, il existe **deux types de vecteurs  $(q_i, d_i)$ , pour lesquels le calcul des probabilités conditionnelles est indispensable :**

- 1- Les vecteurs  $(q_i, -1)$  où  $q_i$  est un état "0" ou un état supérieur à 12.

L'état "0" correspond à une partie (fenêtre 3\*3) de l'image où aucun contour n'est détecté, on lui associe donc la direction indéterminée (-1).

La présence d'un état supérieur à 12 (représentant des angles droits) indique la détection de plusieurs directions macroscopiques dans le voisinage du site. La compatibilité de cet état avec une seule orientation est impossible, il est donc compatible uniquement avec la direction -1

## 2- Deux cas sont alors possibles :

- Si une seule direction macroscopique " $d_j$ " apparaît dans le voisinage du site " $i$ ", aucune création de "coin" n'est alors envisageable. Les seuls états " $q_i$ " devant être pris en compte sont les états compatibles avec la direction " $d_j$ ". On propage la direction de la ligne macroscopique en affectant à " $d_i$ " la valeur du site voisin " $d_j$ ". En effet, le but est de prolonger une ligne de contour, ayant une certaine orientation, par des éléments de contour dans cette même direction.
- Si l'on détecte plusieurs directions macroscopiques dans les sites avoisinants, ou si il apparaît plusieurs états voisins non nuls (éventuellement avec une direction macroscopique identique), la création d'un "coin", de direction -1 est très probable. Les vecteurs à prendre en compte (ceux qui ne correspondent pas forcément à une énergie infinie) sont donc formés par :

Les couples  $(q_i, d_i)$  où  $q_i$  sont tous les états compatibles avec chaque direction détectée (propagation de la direction : " $d_i = d_j$ ") et les couples  $(q_i, -1)$  formés de tous les états compris entre 1 et 12 avec la direction -1 (formation de "coin").

### Comment choisir le vecteur $(q_i, d_i)$ le plus adapté au site " $i$ " ?

On sélectionne ensuite, en appliquant l'algorithme de l'échantillonneur de Gibbs [4] (ou l'algorithme "ICM" [1]), un vecteur  $(q_i, d_i)$  définitif pour le site " $i$ ", parmi les  $N$  vecteurs  $((q_{i_k}, d_{i_k}) \ k=1..N$ , où  $q_{i_k}$  détermine l'état et  $d_{i_k}$  la direction) susceptibles de convenir.

Pour cela, on calcule les  $N$  distributions de Gibbs  $P_{i_k}$  correspondant aux vecteurs  $(q_{i_k}, d_{i_k})$ ,  $k$  variant de 1 à  $N$  :

$$P_{i_k} = \exp \left( \frac{-1}{T} \left( \sum_{i,j: ||i-j|| \leq 2} V2(q_{i_k}, d_{i_k}, q_j, d_j) + \sum_i V1(q_{i_k}, d_{i_k}) \right) \right)$$

On normalise ensuite ces " $k$ " probabilités :

$$P\_norma\_i_k = \frac{P_{i_k}}{\sum_{k=1}^N P(e_{i_k})}$$

Pour choisir la nouvelle variable aléatoire pour le site " $i$ ", on effectue un tirage aléatoire :

$$Si \left( \sum_{k=1}^{n-1} P\_norma\_i_k < \alpha \leq \sum_{k=1}^n P\_norma\_i_k \right)$$

où  $\alpha$  est une variable aléatoire de distribution uniforme.

Alors la nouvelle configuration pour l'instant  $t$  est :

$$X_s(t) = X_s(t-1) \text{ pour } s \neq i$$

et

$$X_i(t) = (q_i, d_i)$$

### 3.4 Application pour la restauration d'image de contours.

La donnée initiale représente une image avec des contours incomplets, ce qui signifie l'absence de certains éléments de contours. Par contre, l'image à compléter n'est pas bruitée, on ne détecte donc pas de "faux" contours. La seule information initiale est donc la connaissance d'une matrice binaire qui permet de savoir si pour un site donné, un contour est détecté.

Avant d'appliquer l'algorithme d'optimisation d'énergie décrit précédemment, il faut initialiser pour chaque site " $i$ ", le vecteur  $(q_i, d_i)$  où " $q_i$ " représente l'état et " $d_i$ " la direction, afin de modéliser l'image initiale.

#### 3.4.1 Initialisation des états.

L'ensemble des 21 états prédéfinis est décrit précédemment (figure 2).

La détermination des différents états, pour chaque site  $i$  s'effectue directement à partir de l'image initiale :

Chaque état " $q_i$ " est déterminé à partir de la mise en correspondance d'une fenêtre de taille  $3 \times 3$ , centrée sur le pixel " $i$ " de l'image initiale et des états prédéfinis. Pour chaque site de l'image d'entrée, on génère donc un état. On détermine ainsi une matrice d'éléments de contours.

#### 3.4.2 Initialisation des directions macroscopiques.

L'initialisation de la deuxième composante du champ de Markov, la direction macroscopique est plus compliquée, et elle est décrite de manière très succincte par J.L Marroquin [5] et [6].

La direction d'une ligne de contours est une propriété globale, qui ne peut être initialisée sans propagation d'information.

Cette composante correspond en fait au numéro de direction défini par la première colonne de la table 3.

Cette initialisation s'effectue à partir de la matrice d'états définie précédemment et permet de connaître en chaque site, où une ligne de contours est détectée, sa direction macroscopique. Plus précisément, " $d_i$ " représente le numéro de direction, correspondant au sous intervalle homogène contenant la pente de la droite passant par le site " $i$ ".

### Initialisation des numéros de direction :

Les états 0 (absence de contour) et supérieurs à 12 ("coins") sont associés au numéro de direction -1 (direction indéfinie).

Pour les autres états, on définit le numéro de direction de la manière suivante :

Pour un site donné, on essaie tous les numéros de direction (compris entre 0 et 11 : voir la table 3) compatibles avec l'état du site étudié. On comptabilise alors le nombre d'états, de la fenêtre 3\*3 centrée sur ce site, compatibles avec chacun de ces numéros de direction. Le choix d'un numéro définitif doit alors être effectué.

Pour cela, on réitère l'algorithme décrit ci-dessous, implanté de façon parallèle (le traitement de tous les sites de la matrice s'effectue simultanément), jusqu'à ce qu'on ne puisse plus affecter de numéro de direction aux sites non initialisés.

Dans cet algorithme, plusieurs cas doivent être pris en compte :

- Si aucun état voisin n'est compatible avec les numéros de direction essayés, c'est-à-dire si un état non nul est isolé (tous les états des sites voisins sont nuls), le numéro de direction affecté en ce site est -1. On suppose pour cela, qu'une ligne de contour associée à une direction est constituée au moins deux éléments de contour.

Exemple :

Fenêtre 3\*3  
de la matrice des états  
centrée sur le site " $i$ "

0	0	0
0	5	0
0	0	0

Fenêtre 3\*3  
de la matrice des numéros  
de direction correspondante

-1	-1	-1
-1	-1	-1
-1	-1	-1

- Si une unique direction est compatible avec l'état du site étudié " $i$ " et les états voisins, cette direction est définitive pour le site " $i$ ".

Exemple :

Etats :

0	0	0	0
0	4	0	0
0	0	12	0
0	0	0	0

Numéros  
de direction :

-1	-1	-1	-1
-1	10	-1	-1
-1	-1	10	-1
-1	-1	-1	-1

En effet, les numéros de direction 8, 9, 10 sont compatibles avec l'état 4, et les numéros 10, 11 avec l'état 12. La direction 10 est donc choisie pour initialiser les deux sites voisins.

- Si plusieurs numéros de direction proposés sont compatibles avec au moins un état voisin, le numéro sélectionné pour le site "i" est celui qui a le plus grand nombre d'états voisins compatibles, ou le numéro définitif d'un site voisin compatible. Ainsi, on propage la direction de la ligne de contour.

Exemple :

Etats :

0	0	0	0	0
0	11	2	2	0
0	0	0	0	0

Numéros  
de direction :

-1	-1	-1	-1	-1
-1	1	1	1	-1
-1	-1	-1	-1	-1

Numéros de direction compatibles avec :

- L'état 11 : 1, 2.
- L'état 2 : 0, 1, 11.

Le site de l'état 11 possède une direction définitive 1. Le numéro de direction du site voisin correspondant à l'état 2 est affecté avec ce même numéro. On propage cette direction pour le dernier site.

- Si plusieurs sites voisins ont un numéro de direction définitif, un tirage aléatoire parmi ceux qui sont compatibles avec l'état de site "i" ("qi") est nécessaire.



Exemple :

Etats :

0	0	0	0	0
0	4	0	0	0
0	0	12	2	0
0	0	0	0	0

Numéros  
de direction :

-1	-1	-1	-1	-1
-1	10	-1	-1	-1
-1	-1	10	11	-1
-1	-1	-1	-1	-1

Numéros de direction compatibles avec :

- L'état 4 : 8, 9, 10
- L'état 12 : 10, 11
- L'état 2 : 0, 1, 11

Les sites correspondants aux états 4 et 2 ont une direction définitive : 10 et 11, car ce sont les seules directions compatibles avec l'état du site et au moins un état voisin.

Le numéro de direction du site, ayant pour état 12, est choisi aléatoirement parmi les numéros 10 et 11, car ces deux directions ont le même nombre d'états voisins compatibles.

Après plusieurs réitérations de cet algorithme, le traitement des sites n'ayant pas été initialisé nécessite également l'étude de plusieurs cas possibles :

- Si trois numéros de direction sont compatibles avec l'état ("q<sub>i</sub>") du site à initialiser, ceci implique que l'état "q<sub>i</sub>" est forcément un état compris entre 1 et 4 (voir la table 3). Dans ce cas, si l'état du site est l'état 1, le numéro de direction 6 est affecté au site "i" (voir la table 3), de même l'état 2 correspond au numéro de direction 0, l'état 3 au numéro 3 et l'état 4 au numéro 9.

Exemple :

Etats :

0	0	0	0
0	2	2	0
0	0	0	0

Numéros  
de direction :

-1	-1	-1	-1
-1	0	0	-1
-1	-1	-1	-1

L'état 2 est compatible avec les numéros de direction 0, 1, 11. On privilégie dans ce cas la direction 0, afin de prolonger cette ligne de manière horizontale (avec des états droits et non pas avec des éléments 10, 11, 9 ou 12).

- Si le nombre de numéros de direction compatibles avec l'état " $q_i$ " est égal à 2, un tirage aléatoire est effectué afin de choisir, le premier ou le deuxième numéro, et de l'affecter à tous les sites de ce type.

Exemple :

Etats :

0	0	0	0
0	12	9	0
0	0	0	0

Numéros  
de direction :

-1	-1	-1	-1
-1	10	10	-1
-1	-1	-1	-1

Les directions 10 et 11 sont compatibles avec les états 9 et 12. Dans ce cas, aucun critère ne permet de sélectionner une des deux directions, un tirage aléatoire est donc effectué. Si on sélectionne la première direction (ici le numéro 10), ce choix est effectué pour tous les états du même type.

**Remarque :**

Cet algorithme de propagation de directions permet d'obtenir le maximum de sites voisins ayant un numéro de direction identique, ceci pour permettre de prolonger une ligne dans la direction la mieux adaptée. En effet, on remarque que deux sites voisins doivent avoir la même direction, pour que l'énergie  $V_2$  puisse avoir pour valeur  $VC$ . Or ce paramètre permet de prolonger les lignes de contours dans la direction appropriée.

### 3.4.3 Exemple simple de prolongements et de raccourcissements de lignes.

#### 3.4.3.1 Implantation parallèle.

Deux méthodes ont été mises en œuvre dans le but de minimiser l'énergie : l'algorithme de l'échantillonneur de Gibbs [4], et l'algorithme déterministe "ICM" [1].

Les méthodes d'optimisation d'énergie utilisées, effectuent à un instant donné la mise à jour du vecteur  $(q,d)$  d'un site connaissant les vecteurs des sites voisins. Les nouvelles valeurs peuvent donc être sélectionnées en parallèle pour des sites non voisins. L'utilisation d'un champ de Markov d'ordre 2 (8-connexité) permet la remise à jour de tous les sites de l'image, en quatre étapes. Une phase de quatre étapes est appelée **itération**. La première étape ne tient compte que des sites d'abscisse et d'ordonnée paire, la seconde d'abscisse impaire et d'ordonnée paire, etc...

Cette mise en oeuvre parallèle pour une matrice 5\*5 peut être schématisée par :

i\j	0	1	2	3	4
0	1	2	1	2	1
1	4	3	4	3	4
2	1	2	1	2	1
3	4	3	4	3	4
4	1	2	1	2	1

où i représente les ordonnées, j les abscisses et la valeur du pixel (i,j), l'étape traitant ce site (par exemple le site d'abscisse 3 et d'ordonnée 2 est pris en compte au cours de la deuxième étape).

L'algorithme d'optimisation nécessite plus d'une mise à jour par site. Après avoir étudié tous les sites de l'image, on réitère les différentes étapes. Les étapes "i" et "i+4\*k" (avec i et k entiers positifs, et i compris entre 1 et 4) opèrent donc sur les mêmes sites.

### 3.4.3.2 Mise en oeuvre de l'algorithme sur une image de taille réduite.

Dans ses articles [5], [6] J.L Marroquin propose de mettre en oeuvre l'algorithme de l'échantillonneur de Gibbs [4] afin de minimiser l'énergie.

Ici, étudions un exemple simple de segments de contours à compléter, en mettant en oeuvre l'algorithme déterministe "ICM" [1].

Soit une image initiale de taille réduite représentée par :

i/j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2																
3																
4										1	1	1	1	1	1	---
5					1											
6					1											
7					1											
8					1											
9					1											
10					---											

Représentation de l'initialisation des champs de Markov :

La première composante représente les états (compris entre 0 et 20), la deuxième les numéros de direction (compris entre -1 et 11).

i/j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2																
3																
4											(2,0)	(2,0)	(2,0)	(2,0)	---	---
5																
6					(1,6)											
7					(1,6)											
8					(1,6)											
9					---											
10					---											

L'initialisation affecte tous les sites de l'image, mais seuls les vecteurs (q,d) différents de (0,-1) sont représentés.

#### a- Première phase : prolongation des lignes de contours.

La première étape, concernant les sites du type (i,j) avec i et j pairs, n'est pas intéressante pour cette application. En effet, la prolongation des deux segments nécessite un changement de l'état des sites (5,4) et (4,9). C'est au cours de la seconde étape que s'effectue la mise à jour du site (4,9).

#### Etude détaillée de la remise à jour du site (4,9) au cours de la deuxième étape :

La nouvelle valeur du site est sélectionnée en fonction des valeurs (q<sub>j</sub>,d<sub>j</sub>) (q<sub>j</sub> : état, d<sub>j</sub> : numéro de direction) des sites voisins représentés ici :

Etats voisins :

i/j	8	9	10
3	0	0	0
4	0	?	2
5	0	0	0

Numéros de direction  
des sites voisins :

i/j	8	9	10
3	-1	-1	-1
4	-1	?	0
5	-1	-1	-1

Notation :

On numérote chaque voisin de la manière suivante :

0	1	2
3		4
5	6	7

Pour minimiser l'énergie (ou maximiser la probabilité) obtenue pour ce site, on applique l'algorithme "ICM" [1] (ou celui de l'échantillonneur de Gibbs [4]) dont le principe est le suivant :

On détermine les probabilités correspondantes à chaque vecteur (q,d) à considérer. La nouvelle mise à jour est effectuée en sélectionnant le vecteur (q,d) associé à la probabilité maximale.

La première colonne du tableau suivant indique tous les couples (q,d) à considérer pour cette nouvelle mise à jour (voir §3.3 : implantation pratique).

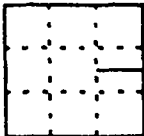
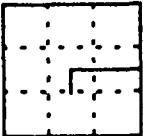
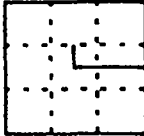
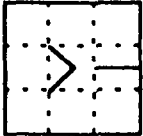
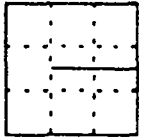
Les connexions entre l'état testé et les huit voisins sont déterminées dans les huit colonnes suivantes, alors que la dernière colonne indique l'énergie totalisée pour le site (4,9) par le vecteur (q,d) essayé.

Dans cet exemple, un unique numéro de direction est déterminé. On considère donc les états 0 et supérieurs à 12 avec la direction -1, ainsi que les états compatibles avec l'unique direction macroscopique 0.

(état, dir.)	Connexion avec le voisin :								Energie	Potentiel de Gibbs
	0	1	2	3	4	5	6	7		
(0,-1)	o	o	o	o	e	o	o	o	VE	$\exp(-VE)$
(13,-1)	o	o	o	o	s	o	e	o	VE+VT	$\exp(-VE-VT)$
(14,-1)	o	o	o	e	t <sub>1</sub>	o	e	o	2VE+VT <sub>1</sub>	$\exp(-2VE-VT_1)$
(15,-1)	o	e	o	o	s	o	o	o	VE+VT	$\exp(-VE-VT)$
(16,-1)	o	e	o	e	t <sub>1</sub>	o	o	o	2VE+VT <sub>1</sub>	$\exp(-2VE-VT_1)$
(17,-1)	o	o	e	o	-	o	o	e	$\infty$	0
(18,-1)	e	o	o	o	t	e	o	o	2VE+VT	$\exp(-2VE-VT)$
(19,-1)	o	o	o	o	-	e	o	e	$\infty$	0
(20,-1)	e	o	e	o	-	o	o	o	$\infty$	0
(2,0)	o	o	o	e	a	o	o	o	VE+VC	$\exp(-VE-VC)$

Voici les configurations correspondantes aux cinq "meilleures" (les plus grandes) probabilités conditionnelles :

### Configurations possibles pour la fenêtre 3\*3 centrée en (4,9)

Configuration :					
Vecteur :	(0,-1)	(13,-1)	(15,-1)	(18,-1)	(2,0)
Energie :	VE	VE+VT	VE+VT	2VE+VT	VE+VC

Si on normalise toutes les probabilités des vecteurs à considérer, on obtient :

$$\begin{aligned}
 \text{Proba\_norma}_1(0,-1) &= C \\
 \text{Proba\_norma}_2(13,-1) &= C \cdot \exp(-VT) \\
 \text{Proba\_norma}_2(14,-1) &= C \cdot \exp(-VE-VT_1) \\
 \text{Proba\_norma}_3(15,-1) &= C \cdot \exp(-VT) \\
 \text{Proba\_norma}_2(16,-1) &= C \cdot \exp(-VE-VT_1) \\
 \text{Proba\_norma}_4(18,-1) &= C \cdot \exp(-VE-VT) \\
 \text{Proba\_norma}_5(2,0) &= C \cdot \exp(-VC)
 \end{aligned}$$

$$\text{où } C = \frac{1}{1 + 2 \cdot \exp(-VT) + \exp(-VE-VT) + \exp(-VC) + 2 \cdot \exp(-VE-VT_1)}$$

Les probabilités associée aux autres vecteurs sont de valeurs nulles.

#### Remarques :

1- Si VC est positif ( $\exp(-VC) < 1$ ), le maximum des probabilités est atteint pour le vecteur (0,-1). On favorise donc l'état 0 par rapport aux autres états. Ceci a pour effet de raccourcir, ou tout au moins de ne pas rallonger (état précédent du site = état 0) les lignes de contours. Inversement, si VC est négatif on favorise le prolongement des lignes.

2- Si VT est négatif et inférieur à VC, VE positif alors on a :

$$\begin{aligned}
 \exp(-VT) &> 1 \\
 \exp(-VT) &> \exp(-VT-VE) \\
 \exp(-VT) &> \exp(-VC).
 \end{aligned}$$

On privilégie donc l'état 13 ou 15 (éléments ayant un angle droit) par rapport à l'état 0 (état "vide"). Ce choix des paramètres n'est pas adapté car il a pour effet de prolonger les lignes de contours, mais cette prolongation n'est pas effectuée dans la direction voulue ("0"). C'est pourquoi il est judicieux de choisir  $|VC| < VT$ .

3- Si  $VT_1$  est inférieur à  $VT$ , on favorise les vecteurs (14,-1) et (16,-1) par rapport aux vecteurs (15,-1) et (16,-1). Ceci n'est pas compatible avec le but fixé : obtenir au bout d'un certains nombres d'itérations des contours, si possible fermés.

4- Le fait de vouloir rallonger les lignes de contours dans une direction donnée (en continuant le plus droit possible) détermine certaines caractéristiques des paramètres  $VE$ ,  $VT$  et  $VC$ .

Ici le meilleur choix pour le site (4,9) est le vecteur (2,0). Pour que ce vecteur corresponde à une énergie minimum, les conditions suivantes doivent être vérifiées :

$$\begin{aligned} VE+VC &< VE \\ VE+VC &< VE+VT \\ 2VE+VT &< 2VE+VT_1 \\ VE+VC &< 2VE+VT \end{aligned}$$

La première condition découle du fait qu'il est préférable de choisir le vecteur (2,0) plutôt que le vecteur (0,-1), si l'on veut rallonger les contours. La deuxième inéquation défavorise les états 13 et 15, la troisième les états 14 et 16, et la dernière l'état 18.

Ceci est vérifié car les paramètres sont choisis de sorte que :

$$|VC| < VT < VT_1 \ll VE$$

En itérant cet algorithme avec des paramètres identiques, d'autres sites sont remis à jour. Par exemple au cours de la quatrième étape, on sélectionne le vecteur (1,6) pour le site (5,4)

Après la neuvième étape, on obtient la configuration suivante :

i/j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2					(1,6) e=9											
3					(1,6) e=8											
4					(1,6) e=5		(2,0) e=9	(2,0) e=6	(2,0) e=5	(2,0) e=2	(2,0)	(2,0)	(2,0)	(2,0)	---	---
5					(1,6) e=4											
6					(1,6)											
7					(1,6)											
8					(1,6)											
9					---											
10					---											

L'étape au cours de laquelle la remise à jour est effectuée est indiquée dans le tableau ("e=..") ainsi que le vecteur (q,d) sélectionné. Les sites sans numéro d'étape correspondent aux données initiales.

Etudions l'évolution de cette configuration au cours de la dixième étape :

**Comment s'effectue la remise à jour du vecteur (q,d) pour le site (4,5)?**

Ce nouveau choix est déterminé en fonction des sites voisins, dont voici les caractéristiques :

Etats voisins :

i/j	4	5	6
3	1	0	0
4	1	?	2
5	1	0	0

Numéros de direction  
des sites voisins :

i/j	4	5	6
3	6	-1	-1
4	6	?	0
5	6	-1	-1



Les vecteurs nécessitant le calcul des probabilités conditionnelles sont des vecteurs du type :

- $(q,-1)$  avec  $0 \leq q \leq 20$ .

Tous les états sont testés avec la direction -1 car on détecte deux directions macroscopiques différentes (0 et 6) dans le voisinage du site (4,5).

- (1,6)
- (2,0)

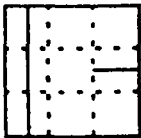
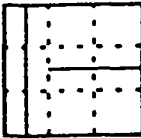
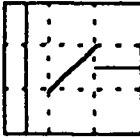
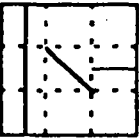
On essaie les deux directions avec leur état compatible.

Voici le tableau indiquant, comme précédemment les connexions de l'état testé avec ses huit voisins, puis pour chaque vecteur  $(q,d)$  l'énergie et la probabilité obtenues.

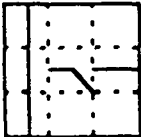
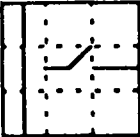
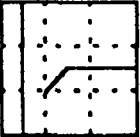
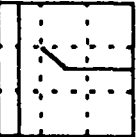
(état, dir.)	Connexion avec le voisin :								Energie	Potentiel de Gibbs
	0	1	2	3	4	5	6	7		
(0,-1)	o	o	o	o	e	o	o	o	VE	$\exp(-VE)$
(1,-1)	o	e	o	-	t	o	e	o	$\infty$	0
(2,-1)	o	o	o	t	a	o	o	o	2VT	$\exp(-2VT)$
(3,-1)	o	o	e	o	t	t	o	o	VE+2VT	$\exp(-VE-2VT)$
(4,-1)	t	o	o	o	t	o	o	e	VE+2VT	$\exp(-VE-2VT)$
(5,-1)	o	o	e	-	t	o	e	o	$\infty$	0
(6,-1)	t	o	o	-	t	o	e	o	$\infty$	0
(7,-1)	o	e	o	-	t	o	o	e	$\infty$	0
(8,-1)	o	e	o	-	t	t	o	o	$\infty$	0
(9,-1)	o	o	o	t	t	o	o	e	VE+2VT	$\exp(-VE-2VT)$
(10,-1)	o	o	e	t	t	o	o	o	VE+2VT	$\exp(-VE-2VT)$
(11,-1)	o	o	o	o	d	t	o	o	2VT	$\exp(-2VT)$
(12,-1)	t	o	o	o	b	o	o	o	2VT	$\exp(-2VT)$
(13,-1)	o	o	o	-	s	o	e	o	$\infty$	0
(14,-1)	o	o	o	-	t <sub>1</sub>	o	e	o	$\infty$	0
(15,-1)	o	e	o	-	s	o	o	o	$\infty$	0
(16,-1)	o	e	o	-	t <sub>1</sub>	o	o	o	$\infty$	0
(17,-1)	o	o	e	-	-	o	o	e	$\infty$	0
(18,-1)	t	o	o	-	t	t	o	o	$\infty$	0
(19,-1)	o	o	o	-	-	t	o	e	$\infty$	0
(20,-1)	t	o	e	-	-	o	o	o	$\infty$	0
(1,6)	o	e	o	-	t	o	e	o	$\infty$	0
(2,0)	o	o	o	t	a	o	o	o	$\infty$	0

Représentations des différentes configurations ayant des probabilités non nulles :

Configurations possibles pour la fenêtre 3\*3  
centrée en (4,5)

Configuration :				
Vecteur :	(0,-1)	(2,-1)	(3,-1)	(4,-1)
Energie :	VE	2VT	VE+2VT	VE+2VT

Configuration :				
Vecteur :	(9,-1)	(10,-1)	(11,-1)	(12,-1)
Energie :	VE+2VT	VE+2VT	2VT	2VT

Ici, l'état "q" choisi établit une "T-jonction" entre la ligne horizontale et la ligne verticale. En effet, comme  $VT \ll VE$ , on a forcément :

$$2VT < VE.$$

Ainsi on choisit l'état 2, 11 ou 12 avec le numéro de direction -1, et non pas l'état 0.

Cette "T-jonction" a une importance capitale dans la phase suivante qui consiste à diminuer les lignes "ouvertes". En effet, on remarquera que la ligne horizontale est définitivement connectée à la ligne verticale et qu'aucun raccourcissement de la ligne (du côté gauche) n'est possible.

Supposons (trois vecteurs ont la même probabilité d'être sélectionnés) que l'on choisisse l'état 2 avec la direction -1, comment évolue alors le système?

Si la phase de prolongement des lignes n'est pas terminée (une période doit être fixée), la ligne verticale continue de se prolonger, alors que la prolongation de la ligne horizontale est stoppée par la "T-jonction".

Si la période de la phase de prolongation est égale à 12, la configuration obtenue après cette étape est la suivante :

i/j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1					(1,6) e=12											
2					(1,6) e=9											
3					(1,6) e=8											
4					(1,6) e=5	(2,-1) e=10	(2,0) e=9	(2,0) e=6	(2,0) e=5	(2,0) e=2	(2,0)	(2,0)	(2,0)	(2,0)	---	---
5					(1,6) e=4											
6					(1,6)											
7					(1,6)											
8					(1,6)											
9					---											
10					---											

#### b- Deuxième phase : raccourcissement des lignes de contours.

L'étape suivante débute une nouvelle phase qui consiste à raccourcir les lignes "ouvertes" (non connectées) de contours. La période durant laquelle cette étape (raccourcissement) est mise en oeuvre est identique à celle de l'étape précédente (rallongement). Lorsque la période est trop courte pour permettre la création de toutes les connexions, certains segments ont la même longueur avant et après ces deux phases. Le fait d'avoir la même période pour les deux étapes permet d'éviter les pertes d'information.

#### Comment s'effectue le raccourcissement des lignes de contours?

Supposons que le site (4,5) ne soit pas modifié après l'étape 14. En fait, comme il a été démontré précédemment, les trois vecteurs : (2,-1), (11,-1), (12,-1) sont équiprobables. Pour ce site, la remise à jour du vecteur (q,d) s'effectue de manière identique dans les deux étapes (car le choix du vecteur est indépendant du signe de la constante VC).

**Le traitement du site (1,4) (dernier site, non connecté, mis à jour pendant la phase précédente) s'effectue au cours de l'étape 16 :**

Etats voisins :

i/j	3	4	5
0	0	0	0
1	0	?	0
2	0	1	0

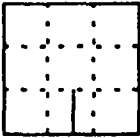
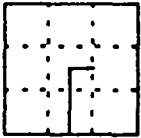
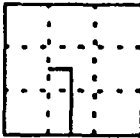
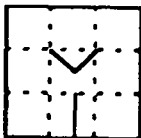
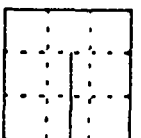
Numéros de direction  
des sites voisins :

i/j	3	4	5
0	-1	-1	-1
1	-1	?	-1
2	-1	6	-1

Les différents vecteurs associés à une probabilité non nulle se déterminent facilement par symétrie, à partir de l'étude effectuée pour le site (4,9).

Vecteurs (q,d) correspondant aux cinq "meilleures" (plus grandes) probabilités strictement positives :

Configurations possibles pour la fenêtre 3\*3  
centrée en (1,4)

Configuration :					
Vecteur :	(0,-1)	(13,-1)	(14,-1)	(20,-1)	(1,6)
Energie :	VE	VE+VT	VE+VT	2VE+VT	VE+VC

Dans cette phase de raccourcissement des lignes de contours non connectées, on a imposé aux paramètres de vérifier :

$$\begin{aligned} &VC > 0 \\ &\text{et} \\ &|VC| < VT < VT_1 \ll VE. \end{aligned}$$

L'énergie minimale est donc atteinte pour le vecteur (0,-1).

De même, au cours de l'étape suivante (i=17) le vecteur (1,6) du site (2,4) est substitué par le vecteur (0,-1)

Etudions la mise à jour du site (3,4) effectuée pendant la vingtième étape :

Etats voisins :

i/j	3	4	5
2	0	0	0
3	0	?	0
4	0	1	2

Numéros de direction  
des sites voisins :

i/j	3	4	5
2	-1	-1	-1
3	-1	?	-1
4	-1	6	-1

Les vecteurs associés à une probabilité non nulle sont les suivants :

Vecteur :

Energie :

(0,-1)	VE
(13,-1)	VE+VT
(14,-1)	VE+VT
(15,-1)	2VE+VT <sub>1</sub>
(16,-1)	2VE+VT <sub>1</sub>
(20,-1)	2VE+VT
(1,6)	VE+VC

Les caractéristiques des paramètres définies précédemment impliquent le choix du vecteur (0,-1).

L'étape suivante (i=21) est intéressante car elle prend en compte le site (4,4) :

Les sites voisins sont caractérisés par les valeurs suivantes :

Etats voisins :

i/j	3	4	5
3	0	0	0
4	0	?	2
5	0	1	0

Numéros de direction  
des sites voisins :

i/j	3	4	5
3	-1	-1	-1
4	-1	?	-1
5	-1	6	-1

Une seule direction macroscopique est détectée dans le voisinage du site, mais il existe deux états voisins non nuls. Les vecteurs à considérer sont donc les vecteurs :

- (q,-1) avec  $0 \leq q \leq 20$ .
- (1,6)

D'après les remarques effectuées au cours de l'étude des cas précédents, on a :

$$\begin{aligned} 2VT &< VE \\ |VC| &< VT < VT_1 \end{aligned}$$

Ces paramètres impliquent donc le choix du vecteur (13,-1) pour le site (4,4). Il est important de constater que le choix n'est pas influencé par le signe du paramètre VC. Ceci permet d'éviter la rupture des deux jonctions au cours des étapes suivantes. La formation des angles permet donc d'atteindre un certain équilibre.

Après l'étape 21, un angle droit est donc formé et stabilise la longueur (ici un seul côté a été étudié) des deux segments.

Voici la configuration obtenue :

i/j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2																
3																
4					(13,-1)	(2,-1)	(2,0)	(2,0)	(2,0)	(2,0)	(2,0)	(2,0)	(2,0)	(2,0)	---	---
5					(1,6)											
6					(1,6)											
7					(1,6)											
8					(1,6)											
9					---											
10					---											

Représentation de l'évolution de l'échantillon de l'image initiale :

i/j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2																
3																
4					1	1	1	1	1	1	1	1	1	1	1	---
5					1											
6					1											
7					1											
8					1											
9					1											
10					---											

### c- Récapitulation :

Pour restaurer une image de contours incomplets, l'algorithme mis en oeuvre se décompose en une succession d'étapes. Chaque étape est constituée de deux phases distinctes : La première d'entre elles permet la prolongation des lignes de contours, alors que la seconde les raccourcit jusqu'à obtenir des "T-jonctions". Ces deux phases sont gérées par trois paramètres : VE, VT et VC intervenant dans la fonction d'énergie. Ces trois constantes doivent vérifier la propriété :

$$| VC | < VT < VT_1 \ll VE$$

VE et VT sont donc deux constantes positives, alors que le signe du troisième paramètre VC varie au cours de l'algorithme :

Conditions supplémentaires :

pour rallonger les lignes :  $VC < 0$   
pour les raccourcir :  $VC > 0$

### d- Remarques :

On a constaté, lors de la remise à jour du site (4,5), que trois vecteurs : (2,-1), (11,-1) et (12,-1) ont la même probabilité d'être sélectionné. Dans l'étude précédente, on suppose que le vecteur (2,-1) est choisi.

En choisissant le vecteur (11,-1), on modifie le résultat final :

En effet, la phase de raccourcissement permet de substituer les vecteurs (1,6) des sites (3,4) et (4,4) par des vecteurs du type (0,-1).

On met ensuite à jour le vecteur de site (5,4). Or, lorsqu'il existe plusieurs directions macroscopiques, ou plusieurs sites voisins ayant comme vecteur : (q,-1) avec q un état non nul (contrairement à ce qui est fait par J.L Marroquin [5], [6] ), on considère les vecteurs (q,-1) avec q compris entre 0 et 20 et le vecteur (1,6). L'état 5 peut donc être sélectionné. Cela permet d'obtenir directement une stabilité car deux "T-jonctions" sont créées. Mais dans ce cas, le contour obtenu ne forme pas "exactement" un angle droit :

i/j	3	4	5	6
3				
4			1	1
5		1		
6		1		
7		1		

Ces remarques restent valables si l'on choisit le vecteur (12,-1) pour le site (5,4).

### 3.4.4 Période de phases de prolongements et de raccourcissements des lignes.

Comme nous l'avons vu dans les chapitres précédents, les paramètres VE, VT, VT<sub>1</sub> et VC permettent de configurer l'algorithme de manière à ce que les lignes de contours soient rallongées ou raccourcies. L'obtention d'une configuration stable (contours fermés et entièrement complétés) nécessite des phases successives de prolongation et de raccourcissement de ligne. On appelle **période** le nombre d'itérations pendant lesquelles chacune des phases est mise en œuvre.

Plusieurs méthodes sont envisageables pour définir la période :

#### 3.4.4.1 Période stable.

Un nombre "p" d'itérations est fixé initialement et reste identique pendant la durée du traitement. L'initialisation de la période dépend de l'image à traiter. C'est la taille (en pixels) des parties de contours manquants qui détermine le choix de la période.

Exemple :

Si la plus grande partie manquante du contour est de 15 pixels, la période choisie doit être supérieure à 15/2 pour un algorithme déterministe. En effet, la prolongation d'une ligne est de deux sites par itération.

Pour un algorithme non déterministe (échantillonneur de Gibbs [4]), la période doit être plus grande, car il est possible que des lignes à compléter restent stables lors de certaines itérations (voir paragraphe 3.4.5).

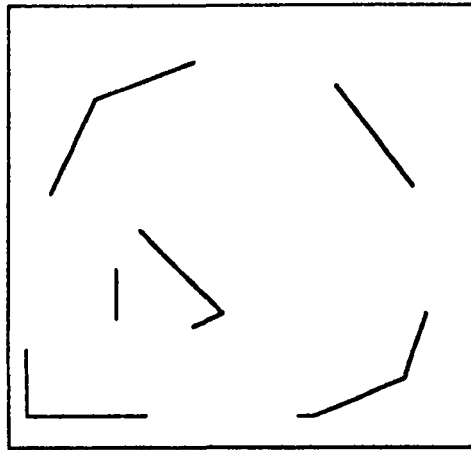
Cette méthode est valable pour la **reconstitution de contours dont les parties manquantes sont homogènes** (approximativement de mêmes tailles).

#### 3.4.4.2 Période variable.

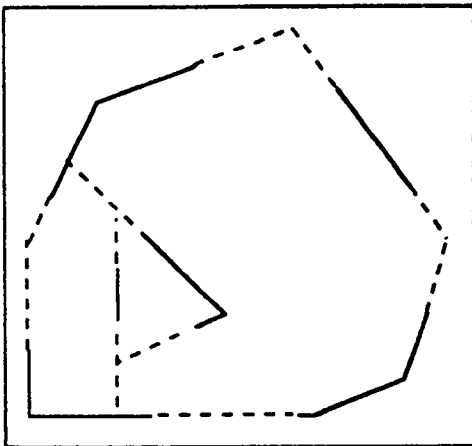
Si les parties à compléter sont de tailles très différentes, un autre algorithme doit être mis en œuvre. Cet algorithme fait intervenir une variation dynamique de la période au cours du traitement. **Au départ, la période fixée est petite**, ceci de manière à privilégier les connexions des segments les plus proches les uns des autres. Petit à petit, **la période est augmentée** jusqu'à atteindre une valeur qui permette de compléter les plus grandes parties manquantes.



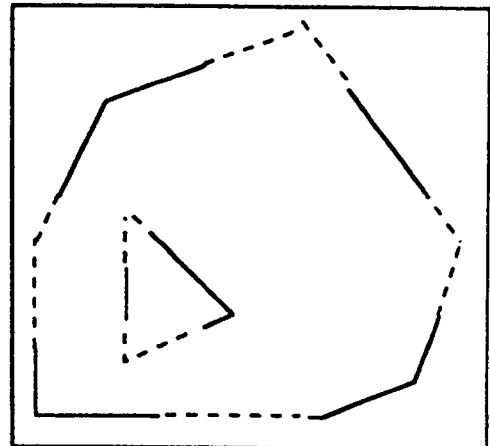
#### 3.4.4.3 Application des deux méthodes sur un exemple simple.



Contour Initial



Algorithme à période fixe



Algorithme à période variable

Contrairement à la méthode de la période variable, celle de la période stable établit des connexions avec des parties de contours éloignées.

#### 3.4.4.4 Problèmes rencontrés.

Comme nous l'avons vu, les deux phases de prolongation et de raccourcissement de lignes sont effectuées avec la même période. Le fait d'avoir une période de prolongation supérieure à celle de la phase de raccourcissement aurait pour conséquence l'instabilité des contours. En effet, certaines lignes seraient rallongées indéfiniment.

Inversement si la phase de raccourcissement est plus longue que celle de prolongation, des segments de contours disparaîtraient au bout d'un certain nombre d'itérations.

#### a. Prolongements des lignes près du bord de l'image.

- Si l'on considère un segment situé près du bord de l'image, on constate que sa prolongation peut être stoppée par la limite de l'image (ceci équivaut à une phase de raccourcissement plus courte). Le segment initial peut donc entièrement disparaître pendant la phase de raccourcissement qui suit. Pour palier à ce problème, **une solution est de considérer l'image comme un tore** (méthode proposée par J.L Marroquin [5], [6] ). Pendant la première phase, le segment est alors prolongé sur le côté opposé de l'image.

Remarque :

Une connexion peut intervenir avec un segment de contour du côté opposé.

- Une autre possibilité est d'insérer l'image à traiter (image1) au centre d'une autre image (image2). Ainsi les sites situés sur le pourtour de l'image2, correspondent à des états nuls de directions indéterminées (-1). En prenant la dimension de la deuxième image suffisamment grande par rapport à la dimension de l'image initiale ( elle doit être proportionnelle à la période choisie), il est possible de supprimer le problème de prolongations de contours sur le côté opposé, existant dans le cas de modélisation d'image par un tore. Le problème est cette fois-ci, un problème de temps de calculs.

- Nous avons expérimenté une autre solution, qui consiste à "bloquer" les lignes au bord de l'image.

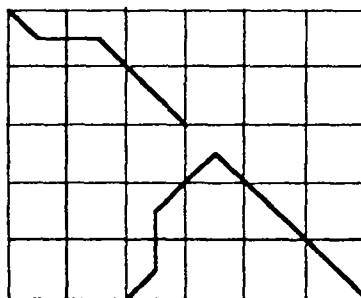
En fait, lorsque dans l'image initiale on détecte une ligne dont un côté est en contact avec le bord de l'image, on empêche alors toutes variations de longueur du côté en question.

Lorsque ce contact s'effectue pendant une phase de prolongation, on empêche alors le rallongement de la ligne tout en comptant le nombre (noté nb) de pixels qui auraient pu être modifiés dans une image plus grande. Ainsi, pendant la deuxième phase, celle des raccourcissement, on attend un nombre (nb) d'itérations avant d'autoriser la diminution du côté de la ligne étudiée.

#### b. Raccord de deux parties de contours.

Exemple :

Considérons la partie de l'image initiale suivante :



La configuration d'états correspondante est la suivante :

i/j	0	1	2	3	4	5
0	12	9				
1			4			
2				19		
3			5		3	
4			8			3

Les directions associées sont définies par le tableau suivant :

i/j	0	1	2	3	4	5
0	11	11				
1			-1			
2				-1		
3			5		4	
4			5			4

**Remarque :**

Notons qu'il est tout à fait possible que le site (1,2) ait pour vecteur, le vecteur (4,-1) même si une unique direction macroscopique est détectée. En effet, si le vecteur (19,-1) sélectionné pour le site (2,3), remplace le vecteur (8,4) après la mise à jour du site (1,2), alors le choix du vecteur (4,-1) est effectué de la manière suivante :

Comme plusieurs directions macroscopiques sont alors présentes dans le voisinage du site (1,2) (directions 11 et 4), les vecteurs (q,-1) avec q compris entre 0 et 20 sont à considérer. C'est le vecteur (4,-1) qui correspond à la probabilité maximale.

**Problème :**

Ici le problème est le **raccordement de trois segments en un point donné.**

Le site a étudié est le site de coordonnées (1,2).

Etats voisins :

i/j	1	2	3
0	9	0	0
1	0	?	0
2	0	0	19

Numéros de direction  
des sites voisins :

i/j	1	2	3
0	11	-1	-1
1	-1	?	-1
2	-1	-1	-1

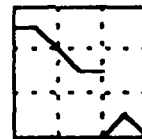
D'après les articles de J.L Marroquin [5], [6] comme une seule direction macroscopique est présente dans le voisinage du site (car le vecteur (19,-1) a été sélectionné pour le site (2,3)), les vecteurs (q,d) à considérer pour ce site sont de la forme :

(q,-1) où q représente l'état nul, ou un état supérieur à 12.

(q,11) où 11 est la direction macroscopique à propager et q les états compatibles avec cette direction (états : 2, 9, 12).

Le vecteur correspondant à la densité de probabilité maximale est le vecteur (12,11) :

Configuration :



Vecteur : (12,11)

Energie : VE + VC

L' énergie correspondante au vecteur (12,11) est égale à : "VE+VC". Comme elle dépend de VC, ceci implique que cette configuration n'est pas stable, car aucune "T-jonction" n'a été établie. En effet, au cours de la phase suivante de raccourcissement ( $VC > 0$ ), le vecteur sélectionné sera le vecteur (0,-1) correspondant à une énergie égale à VE. L'état 12 sera donc remplacé par l'état 0, et un raccourcissement de la ligne aura lieu. Ceci équivaut au problème survenant lorsqu'on prolonge un segment près du bord de l'image (disparition possible du segment lors de la phase de raccourcissement et instabilité du contour).

Une modification a été apporté à la méthode de Marroquin [5], [6] Elle permet d'obtenir une connexion (plus précisément une "T-jonction") lors d'une telle configuration et par conséquent la stabilité du contour.

Lorsque dans le voisinage du site plusieurs directions macroscopiques sont détectées ou lorsqu'il en existe une unique et un état voisin supérieur à zéro avec une direction -1, les vecteurs à considérer sont les suivants :

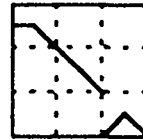
- $(q,-1)$  avec  $q$  compris entre 0 et 20
- $(q,d0)$  où  $d0$  est la direction macroscopique et  $q$  les états compatibles avec cette direction.

Ici les vecteurs supplémentaires à tester pour le site en question sont les vecteurs  $(q,-1)$  avec  $q$  compris entre 1 et 12.

En prenant en compte cette modification, le vecteur sélectionné est le vecteur  $(4,-1)$ .

Les résultats obtenus pour l'état 4 sont les suivants :

Configuration :



Vecteur :  $(4,-1)$

Energie :  $2VT$

L'énergie correspondant au vecteur  $(4,-1)$  ne fait pas intervenir le paramètre VC, mais uniquement le paramètre VT. Le fait de sélectionner ce vecteur permet de créer une "T-jonction" qui est conservée lors de la phase suivante (raccourcissement). Cette configuration est donc stable, et elle évite toutes pertes d'information.

### 3.4.5 Comparaison de deux algorithmes : Echantillonneur de Gibbs et ICM.

J.L Marroquin propose dans ses articles [5] et [6] de mettre en oeuvre un algorithme stochastique : l'échantillonneur de Gibbs [4], afin de trouver le minimum de la fonction d'énergie.

Pour la restauration d'image de contours, l'algorithme déterministe : ICM [1] donne également des résultats convenables. En effet, l'inconvénient de cette méthode est qu'elle ne converge en général pas vers un maximum global, mais local. La qualité du résultat obtenu dépend des conditions initiales. Ici, comme l'image initiale n'est pas bruitée, c'est-à-dire qu'elle ne contient pas de fausse information, on peut mettre en oeuvre cet algorithme.

Dans le cas de mise en oeuvre de l'algorithme de l'échantillonneur de Gibbs [4], le vecteur  $(q,d)$  choisi pour la mise à jour d'un site donné, ne correspond pas

forcément à la densité de probabilité maximale. En effet, comme nous l'avons remarqué précédemment ( voir § 4.3.1.2), pour sélectionner les nouvelles valeurs (état et direction), un tirage aléatoire est effectué.

Supposons que durant la phase de prolongation des lignes, l'état choisi (aléatoirement) soit l'état 0 alors qu'il existe un vecteur (q,d0) (où q représente un état non nul compatible avec la direction d0 de la ligne à prolonger) correspondant à l'énergie minimale.

Supposons également que pendant la phase suivante (de raccourcissement des lignes) tous les vecteurs sélectionnés soient bien ceux qui minimisent l'énergie.

Alors ceci équivaut à une **période de phase de prolongation inférieure à celle de phase de raccourcissement**. Par conséquent, une perte d'information (disparition de contours) peut avoir lieu pendant la phase de raccourcissement (voir § 3.4.4.4).

Les résultats obtenus à partir de la mise en oeuvre de l'algorithme de l'échantillonneur de Gibbs [4] sont très sensibles aux valeurs des paramètres VE, VT, VT<sub>1</sub> et VC choisies. En effet, même si la propriété nécessaire :  $|VC| < VT < VT_1 < VE$  est vérifiée, des différences peuvent être constatées suivant les valeurs des trois constantes.

Les valeurs des paramètres suivants donnent de bons résultats.

Paramètres utilisés pendant la phase de prolongation :

$$VE = 20 ; VC = -4 ; VT = 6 ; VT_1 = 8$$

et pendant la phase de raccourcissement :

$$VE = 20 ; VC = 4,5 ; VT = 6 ; VT_1 = 8$$

Exemple :

Soit l'exemple étudié précédemment (voir §3.4.3.2 étude du site (4,9)):

Etats voisins :

i/j	8	9	10
3	0	0	0
4	0	?	2
5	0	0	0

Numéros de direction  
des sites voisins :

i/j	8	9	10
3	-1	-1	-1
4	-1	?	0
5	-1	-1	-1

En prenant les paramètres :  $VE = 20$  ;  $VC = -4$  ;  $VT = 5$  ;  $VT_1 = 7$  , pour la phase de prolongation des lignes, nous obtenons les valeurs des probabilités normalisées suivantes (voir §3.4.3.2 étude du site (4,9)) :

$Proba\_norma_1(0,-1)$	$= C$	$\approx 0,018$
$Proba\_norma_2(13,-1)$	$= C \cdot \exp(-VT)$	$\approx 0,000$
$Proba\_norma_3(15,-1)$	$= C \cdot \exp(-VT)$	$\approx 0,000$
$Proba\_norma_4(18,-1)$	$= C \cdot \exp(-VE-VT)$	$\approx 0,000$
$Proba\_norma_5(2,0)$	$= C \cdot \exp(-VC)$	$\approx 0,982$

$$\text{où } C = \frac{1}{1 + 2 \cdot \exp(-VT) + \exp(-VE-VT) + \exp(-VC) + \exp(-VE-VT_1)}$$

En choisissant ces valeurs de paramètres, nous avons donc **une probabilité non nulle (1,8 %) de raccourcir** (ou de ne pas rallonger) **la ligne pendant la phase de prolongation**.

De même, en choisissant les paramètres :  $VE = 20$  ;  $VC = 4,5$  ;  $VT = 5$  ;  $VT_1 = 7$ , pour la **phase de raccourcissement**, on obtient **une probabilité non nulle (2,4 %) de rallonger les lignes**.

En fait, il est important de toujours prolonger les lignes non connectées pendant la phase de prolongation et de les raccourcir pendant la deuxième phase. Pour cela, on doit toujours choisir le vecteur (q,d) correspondant à la probabilité maximale. Ceci peut s'obtenir par la mise en oeuvre d'un **algorithme déterministe : ICM [1]**.

### 3.4.6 Modifications effectuées dans le but de traiter des images bruitées

Dans le cas d'une image bruitée, on constate d'une part une perte d'information importante lors de la création de la "matrice" correspondant aux états, ainsi que des fausses information (bruit).

En effet, pour chaque site de l'image de départ, on initialise les états en comparant une fenêtre de taille 3\*3 centrée en chacun de ces sites avec les états prédéfinis. Lorsque cette fenêtre ne correspond pas à un état prédéfini, on considère l'état comme étant nul. Lorsque l'image initiale est bruitée, on constate un nombre important de sites ayant un état initial nul, d'où la perte d'information.

Prenons l'exemple :

Soit une image initiale bruitée (image0) représentée par :

	1	1	1	1	1	1
	1	1	1			
	1					
	1					
	1					

Représentation :

“1” représente les contours de l'image initiale et “1” le bruit.

On remarque, dans cet exemple, que du bruit est détecté dans le voisinage du contour initial.

L'initialisation des états (image1) s'effectue de la manière suivante :

	0	0	0	0	2	2
	0	0	0			
	0					
	1					
	1					

Certains contours ne sont donc pas détectés, et le fait de les supprimer peut avoir des conséquences sur la reconstruction finale.

Pour palier en partie à ce problème, on effectue des traitements préalables sur les images à compléter :

1- Le premier traitement consiste à supprimer les chaînes d'états non nuls, constituées par un nombre “insuffisant” d'éléments. Ceci permet d'éliminer le bruit isolé.

2- Un deuxième traitement permet d'initialiser un état en acceptant plusieurs configurations possibles.



### Schéma représentant les états prédéfinis :

Etat 1 :					
Etat 2 :					
Etat 3 :					
Etat 4 :					
Etat 5 :					
Etat 6 :					
Etat 7 :					
Etat 8 :					
Etat 9 :					
Etat 10 :					
Etat 11 :					
Etat 12 :					
Etat 13 :					
Etat 14 :					
Etat 15 :					
Etat 16 :					
Etat 17 :					
Etat 18 :					
Etat 19 :					
Etat 20 :					

Remarque :

Cette définition des différents états est assez arbitraire, car toutes les configurations possibles ne sont pas prises en compte. Par contre, si on ne considère pas les éléments de contours correspondant aux angles aigus, on remarque que les cas où trois pixels de la fenêtre 3\*3 indiquent la présence d'un contour (20 cas : cf figure ci-dessus) sont étudiés, c'est-à-dire associés à des états prédéfinis. On suppose donc avoir, dans la majeure partie des cas, des contours bruités d'épaisseur égale à un. Par contre, on ne peut pas

déterminer un seuil du rapport signal sur bruit au-delà duquel la reconstruction des contours n'est pas satisfaisante, car c'est la position du bruit par rapport aux contours donnés initialement qui est essentielle.

En prenant en compte cette modification, on obtient les états suivants (*image1*) :

	13	0	0	1	1	1
	0	0	0			
	1					
	1					
	1					

Ainsi, on ne perd pas autant d'information, mais ce résultat n'est pas encore satisfaisant.

On effectue donc, un traitement supplémentaire, qui consiste à créer à partir de la matrice des états obtenus (*image1*), une matrice binaire de contours (*image2*). Pour l'obtenir, on définit pour chaque pixel de la matrice des états (*image1*), l'élément de contour (fenêtres 3\*3) associé. Lorsqu'on considère deux états voisins de l'image *image1*, les fenêtres (3\*3) vont se chevaucher. On conserve pour le site *i* la valeur 1 (présence d'un contour) si au moins un pixel des deux fenêtres correspondant à ce site a une valeur 1 et 0 sinon. On considère ensuite, cette image (*image2*) comme étant une image d'entrée non bruitée. On initialise donc à nouveau les états (*image3*).

Dans l'exemple précédent, on associe l'état 13 à la fenêtre 3\*3 suivante :

	1	1
	1	

On obtient donc l'image binaire de contours suivante (*image2*) :

	1	1	1	1	1	1
	1					
	1					
	1					
	1					

Cette image est alors considérée comme l'image initiale, et l'initialisation des contours (image3) donne cette fois-ci des résultats satisfaisants :

	13	1	1	1	1	1
	1					
	1					
	1					
	1					

On remarque que dans la majeure partie des cas, ces traitements sont efficaces, mais parfois l'information reste erronée et la restauration des contours est mauvaise.

### 3.5 Résultats de simulations.

#### 3.5.1 Mise en oeuvre sur la machine à connexions : CM2.

La méthode de restauration d'image de contours incomplets a été implantée sur la machine à connexions (CM2) [11]. Le langage utilisé est le C\* [12], basé sur le C ANSI, il permet la mise en oeuvre d'algorithmes parallèles sur la CM2.

La machine à connexions [11] possède de 8K à 64K processeurs 1 bit ayant 32K octets de mémoire locale chacun.

Les programmes parallèles sont adaptés à la manipulation de grands ensembles de données. Chaque élément de donnée est associé à un processeur. Tous les processeurs effectuent la même opération sur l'ensemble des éléments en même temps.

Si le nombre d'éléments de données est plus important que le nombre de processeurs (et c'est souvent le cas), le système crée des processeurs virtuels en diminuant la mémoire associée à chaque processeur physique. La taille d'un ensemble de données utilisé dans un programme est ainsi illimitée. La "CM" [11] possédant 16K processeurs physiques peut alors en simuler 32K ou 64K, mais le temps nécessaire pour effectuer chaque opération augmente avec le nombre de processeurs virtuels.

Pour le traitement d'images, et plus particulièrement pour le traitement bas niveau (au niveau pixels), les algorithmes utilisés peuvent souvent s'implanter de façon parallèle car tous les pixels nécessitent en général des traitements locaux et identiques. Ce parallélisme est d'autant plus intéressant que les données (pixels) sont nombreuses. L'image est représentée par une variable parallèle dont chaque élément est un pixel. Chaque pixel est associé à un processeur et les interactions locales sont déterminées par des communications inter-processeurs.

### 3.5.2 Résultats sur des images synthétiques non bruitées.

Cette méthode de restauration d'image de contours incomplets a été testée sur des images synthétiques de taille 128\*128.

Ces résultats ont été obtenus avec les paramètres vérifiant la condition :  
 $|VC| < VT < VT_1 \ll VE$ .

Pour cet algorithme, il suffit de fixer :

$VE = 20, VT = 6, VT_1 = 8, VC = \pm 4$  (varie au cours de l'algorithme)

**Notations :**

L'image initiale est notée en gras :

" 1 " : contour présent initialement et inchangé au cours de l'algorithme.

" 0 " : contour initial supprimé par la suite.

Les contours complétés sont notés : "1".

Les pixels sans valeur représentent les sites où aucun contour n'est détecté (valeur 0).

### Image 1 :

Méthode de la période variable :

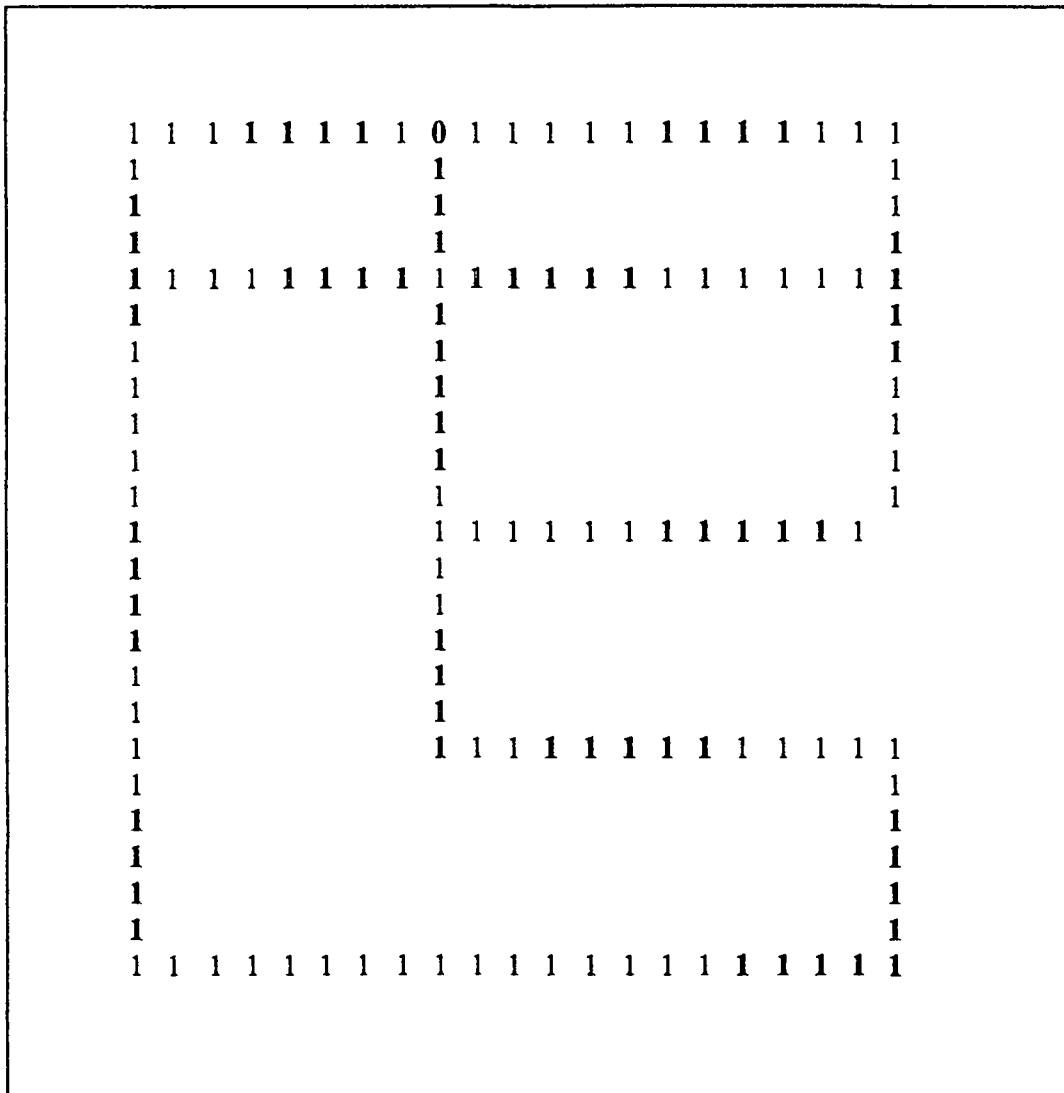
La période initiale est fixée à 3.

Au début de chaque phase de prolongation, on incrémente la période de 7.

Ainsi les phases de prolongation et de raccourcissement sont effectuées avec la même période.

Nombre d'itérations = 2 (2 phases de prolongation et de raccourcissement).

Algorithme utilisé : l'échantillonneur de Gibbs.



### Remarque :

Des jonctions "en croix" ont été déterminées. Elles peuvent être obtenues par exemple, à partir des configurations suivantes :

	1	
2	2	2
	1	

	8	
2	2	2
	1	

	7	
2	2	2
	6	

	1	
1	1	11
	5	

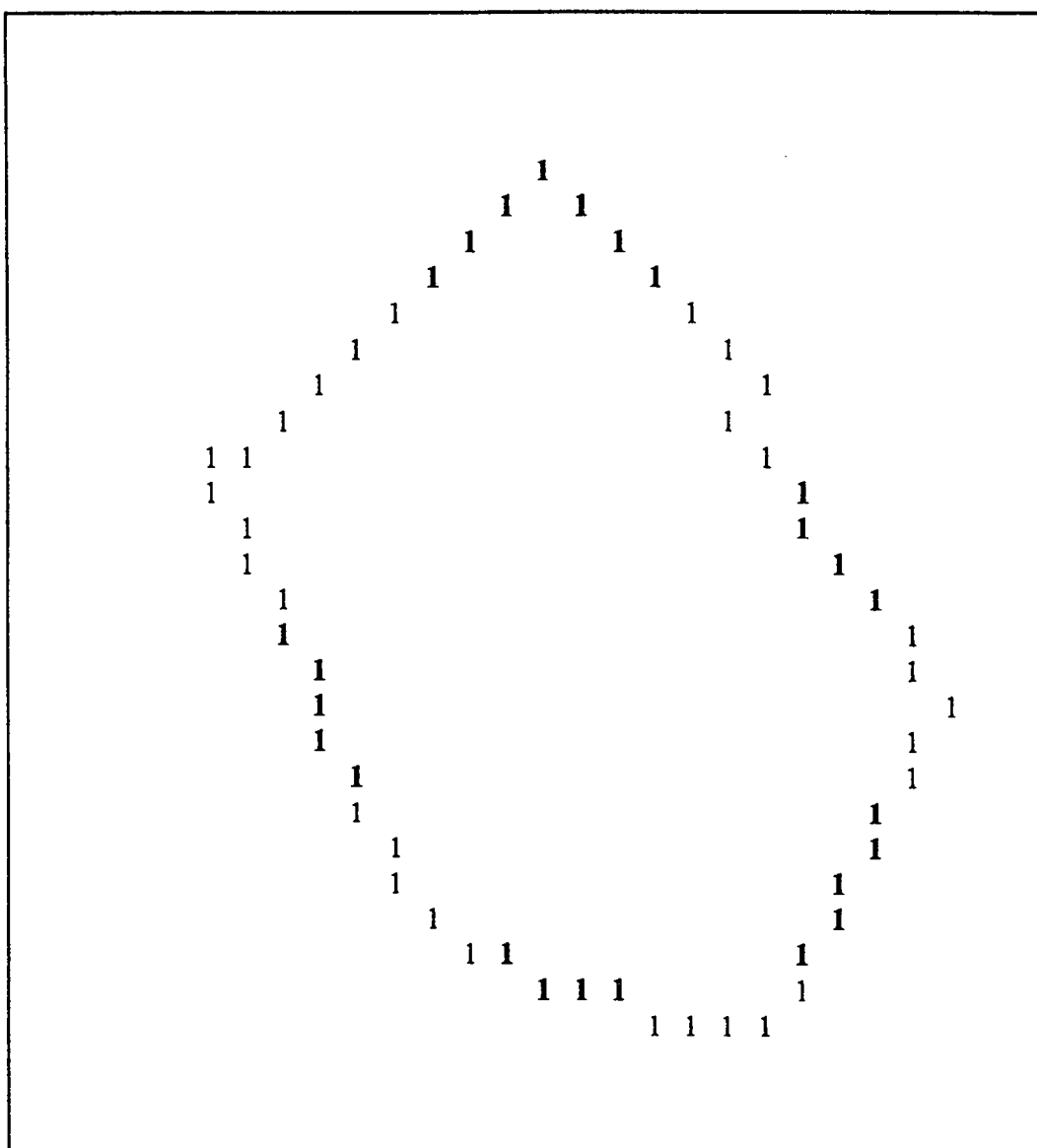
**Image 2 :**

Méthode utilisée :

Période fixe égale à 4.

Nombre d'itérations fixé à 1 (une phase de prolongation et de raccourcissement).

Algorithme mis en oeuvre : l'échantillonneur de Gibbs.



### Image 3 :

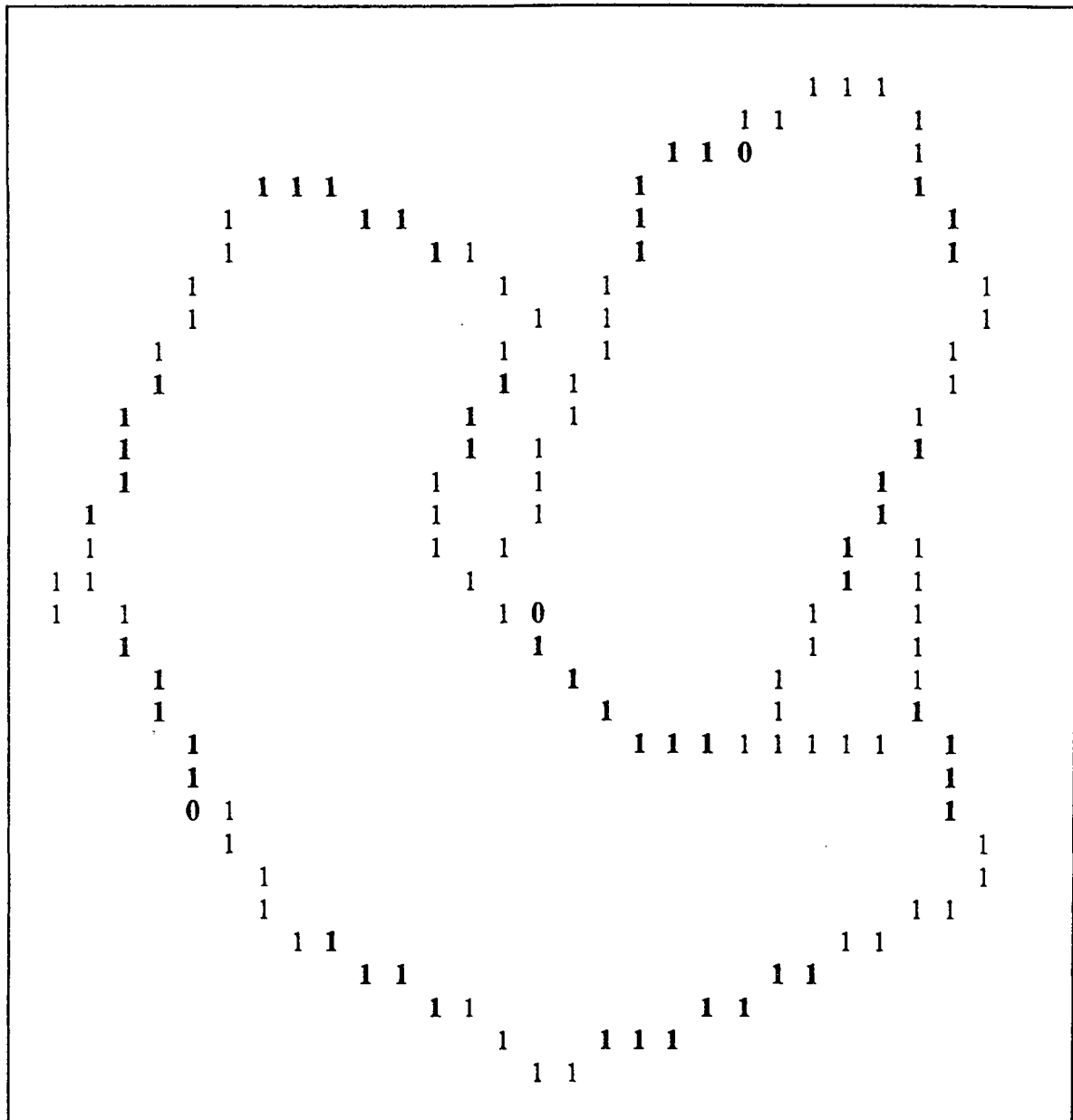
Méthode de la période variable :

Période initiale : 5.

En début de chaque phase de prolongation, on incrémente la période de 7.

Nombre d'itérations : 2 (2 phases de prolongation et de raccourcissement).

Algorithme mis en œuvre : ICM



### 3.5.3 Résultats sur des images synthétiques bruitées.

Notations :

0 : représente du bruit détecté dans l'image initiale qui a été supprimé.

1 : bruit non supprimé.

Image 4 :

Période fixe égale à 4.

Nombre d'itérations fixé à 1 (une phase de prolongation et une de raccourcissement).

Algorithme mis en oeuvre : l'échantillonneur de Gibbs.

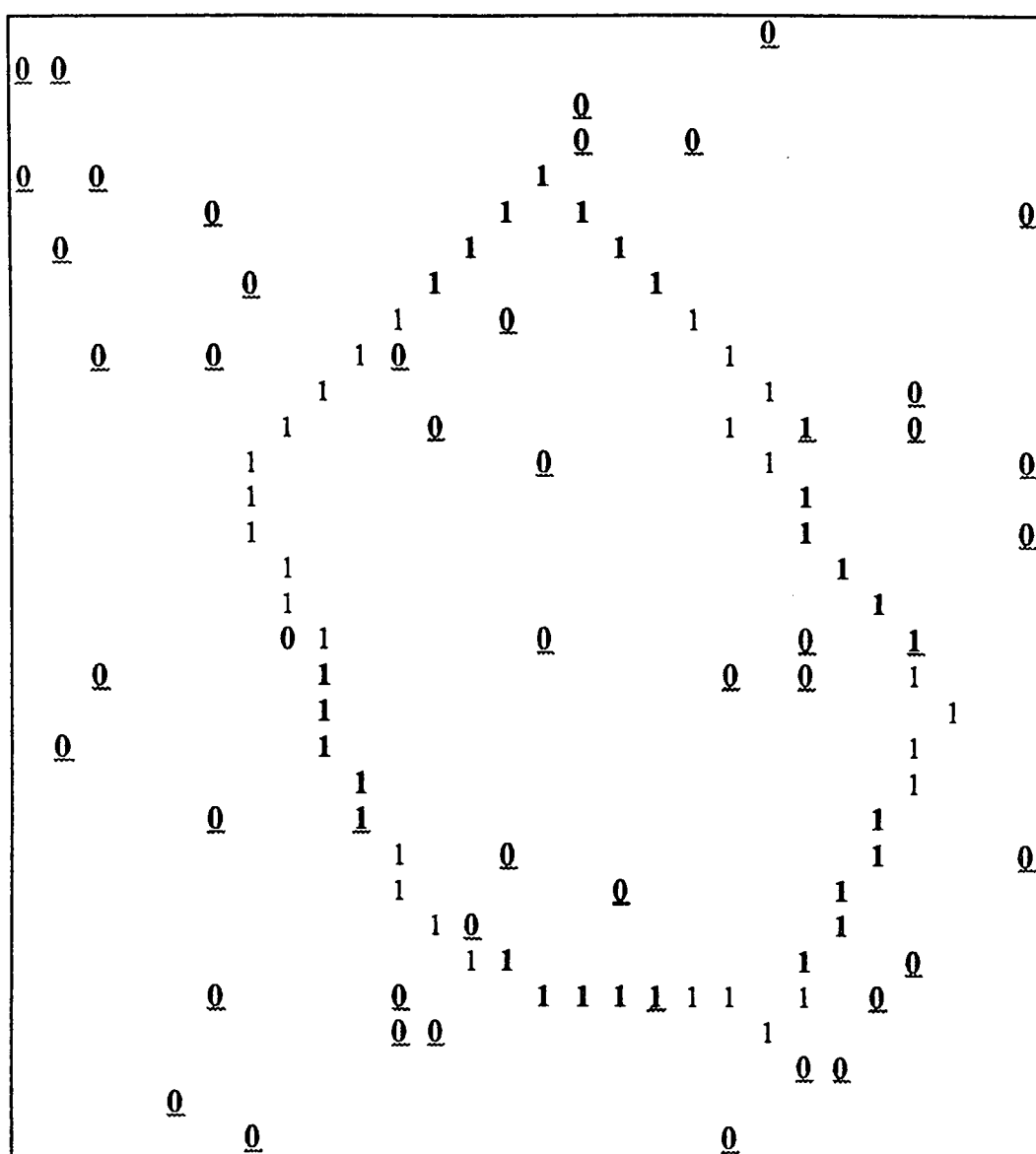




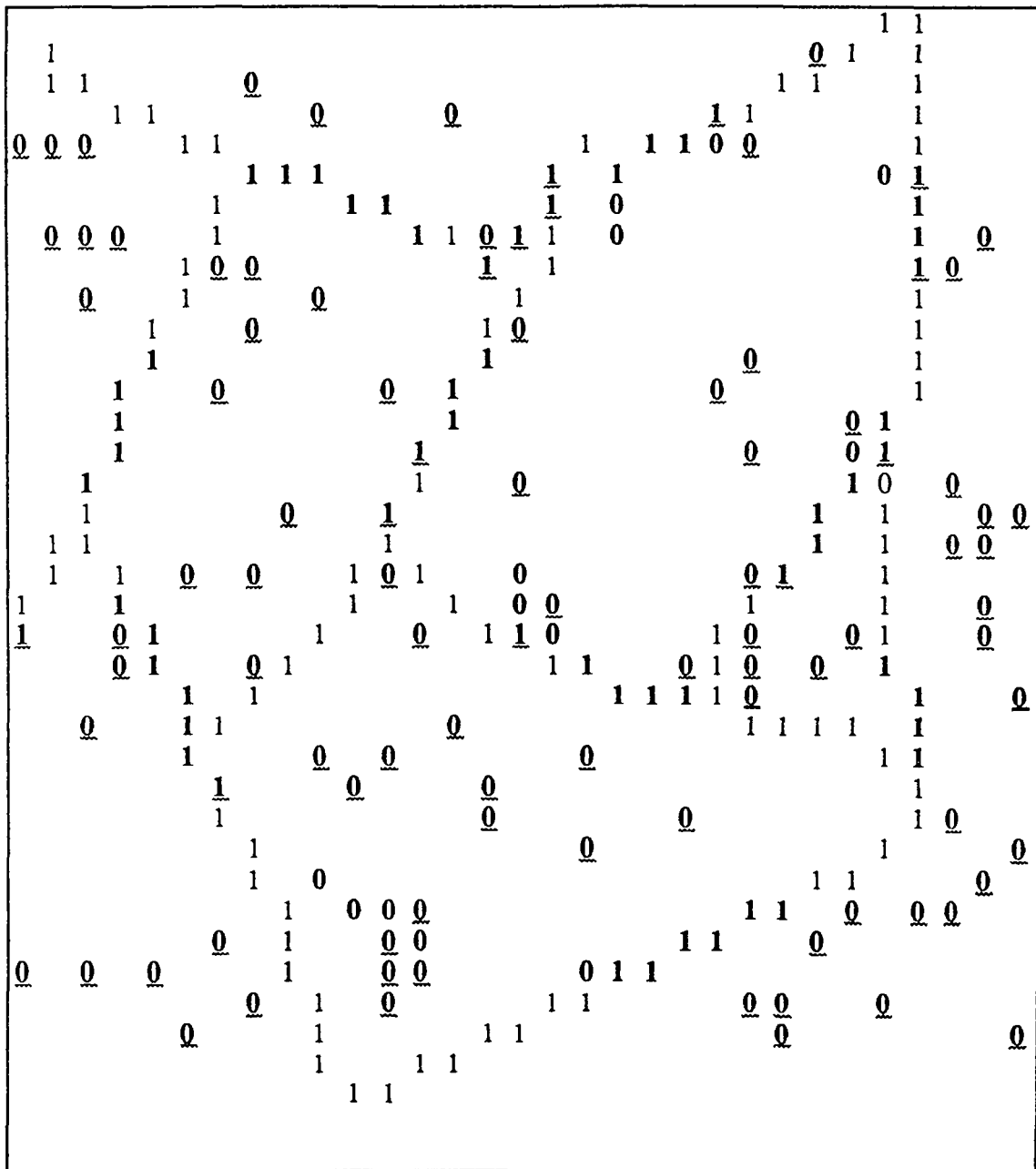
Image 5 :

Méthode de la période fixe :

Période = 12

Nombre d'itérations : 1 (1 phase de prolongation et de raccourcissement).

Algorithme mis en oeuvre : l'échantillonneur de Gibbs.



### **3.5.4 Résultats sur des images de scènes réelles.**

A partir d'une image de scène réelle, on détermine les contours par l'algorithme de Canny-Deriche suivi d'un seuillage des données obtenues. On applique alors l'algorithme décrit précédemment (échantillonneur de Gibbs), dans le but de compléter les contours détectés.

Les images 6, 7 et 8 sont des images représentant un bureau (256\*256). Les images 9, 10, 11, 12 des images satellites (SPOT) (256\*256).

Image 6 :

### Restauration des contours d'une image de scène réelle

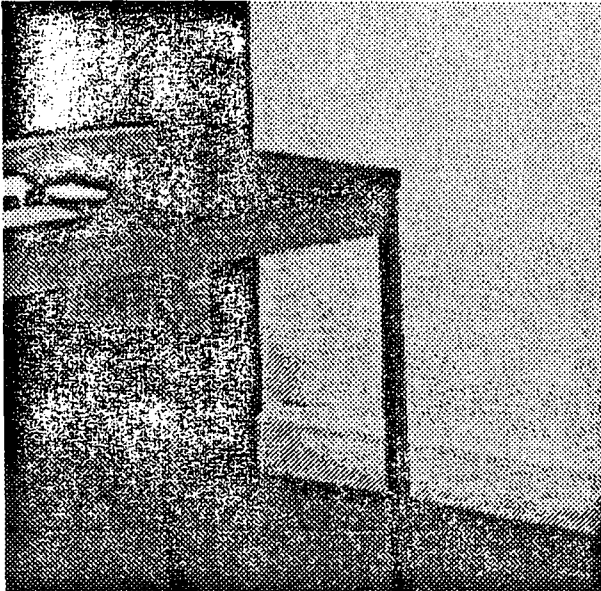


Figure 1 : Image de scène réelle

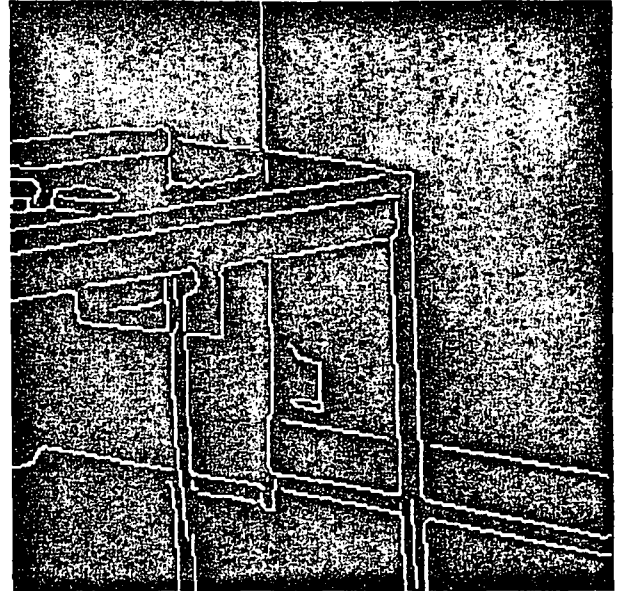


Figure 2 : Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima

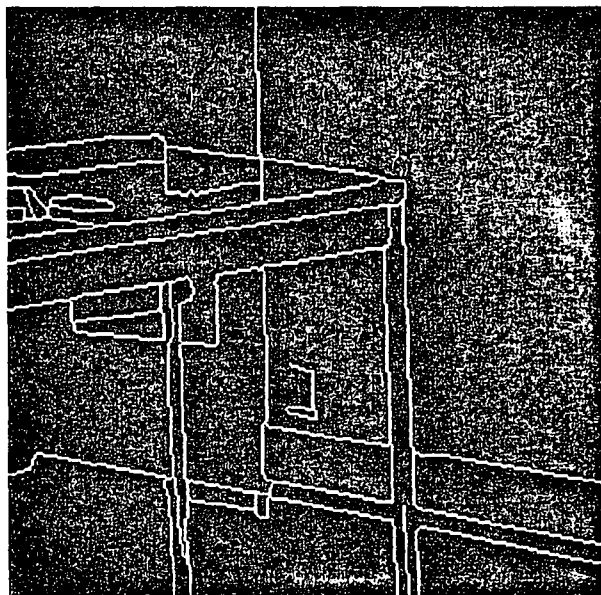


Figure 3 : Restauration des contours incomplets - Période fixe = 12 - Nombre d'itérations = 1 - Algorithme de l'échantillonneur de Gibbs

Image 7 :

### Restauration des contours d'une image de scène réelle

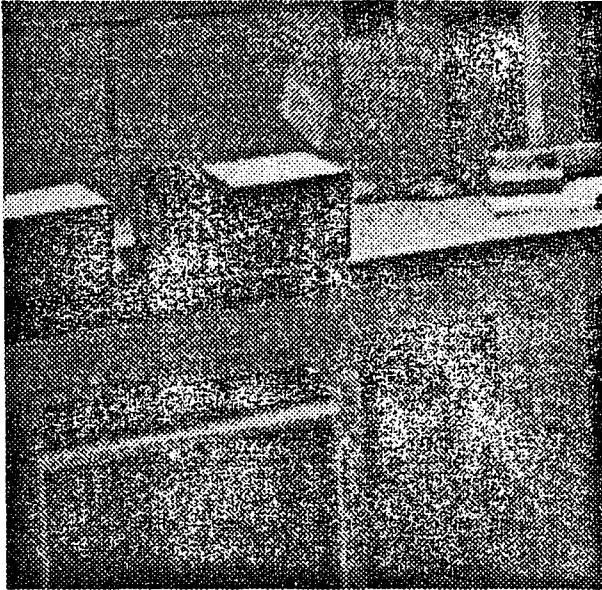


Figure 1 : Image de scène réelle

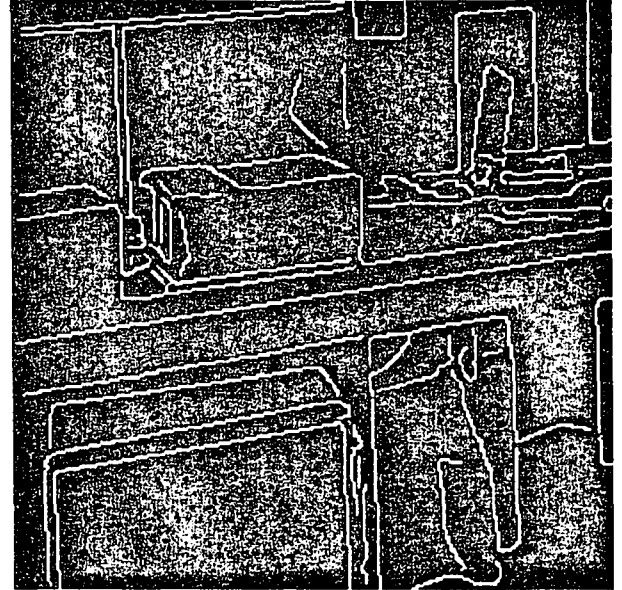


Figure 2 : Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima

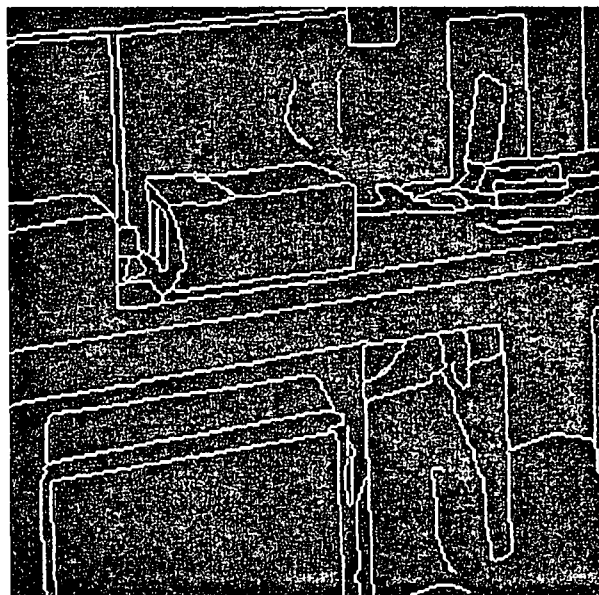


Figure 3 : Restauration des contours incomplets - Période fixe = 5 - Nombre d'itérations = 1 - Algorithme de l'échantillonneur de Gibbs

Image 8 :

### Restauration des contours d'une image de scène réelle

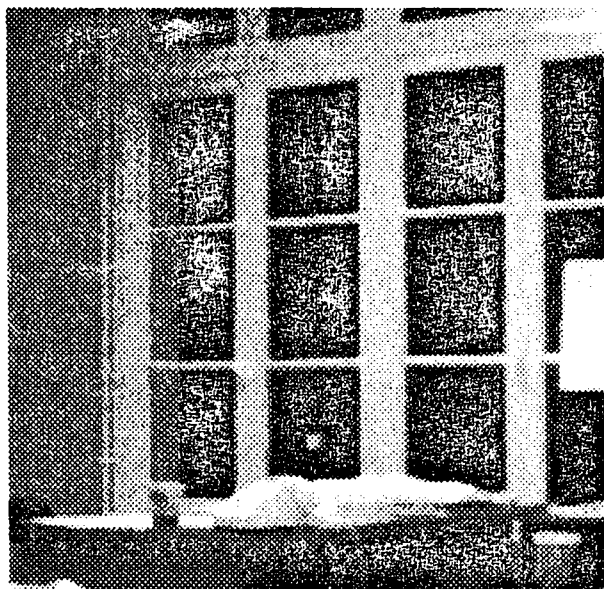


Figure 1 : Image de scène réelle

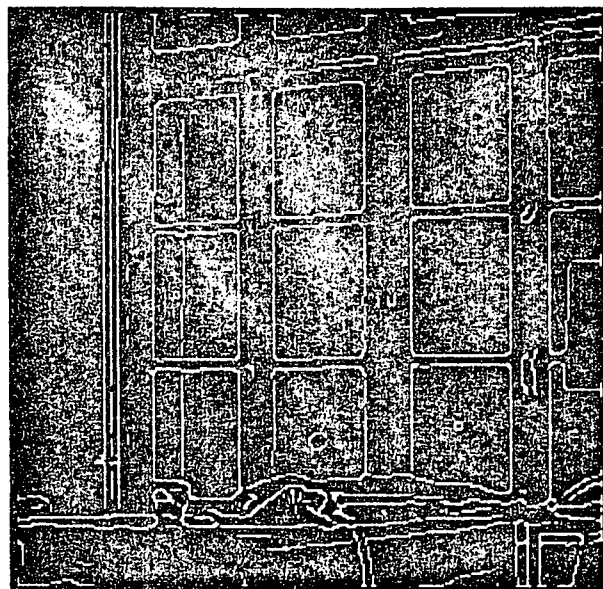


Figure 2 : Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima

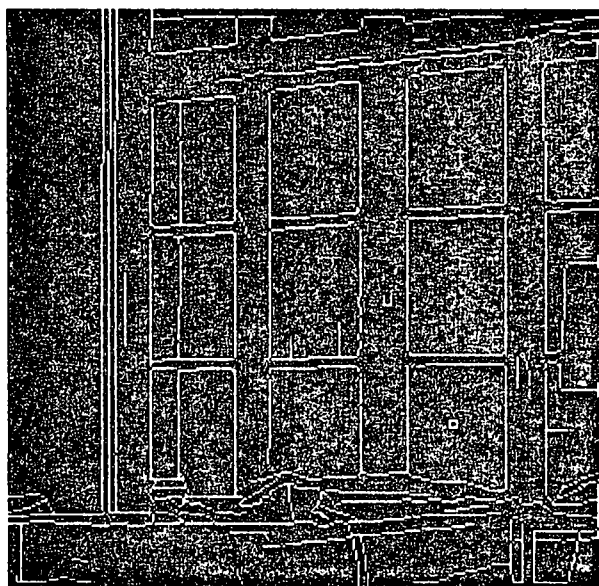


Figure 3 : Restauration des contours incomplets - Période fixe = 7 - Nombre d'itérations = 1 - Algorithme de l'échantillonneur de Gibbs

Image 9 :

### Restauration des contours d'une image satellite



Figure 1 : Image de scène réelle

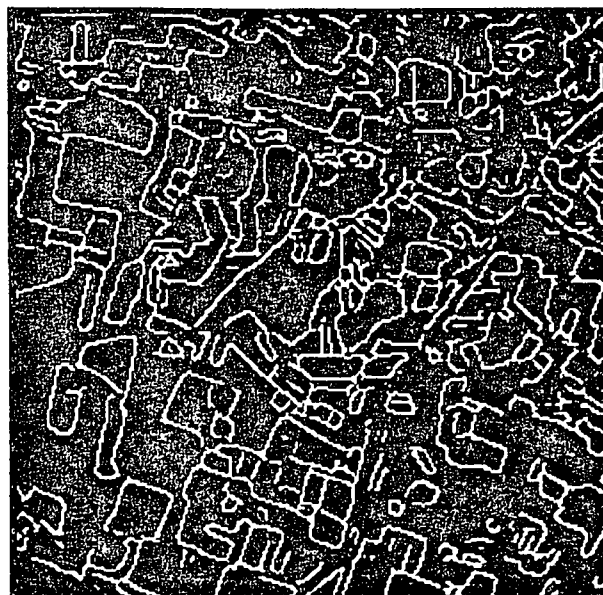


Figure 2 : Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima

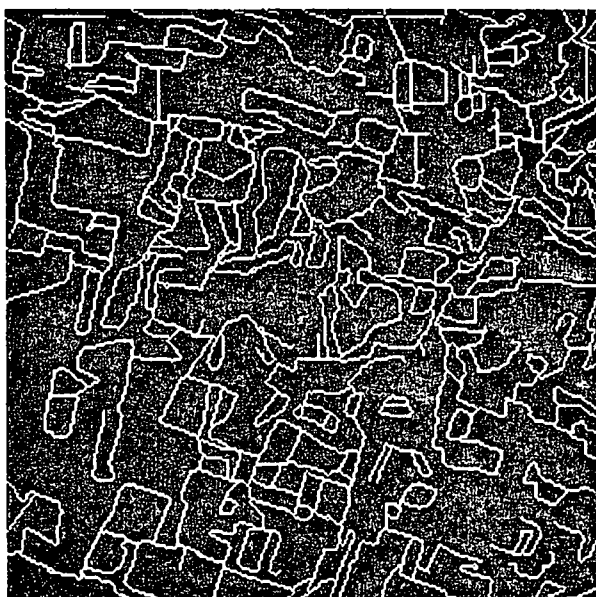


Figure 3 : Restauration des contours incomplets - Période variable = 3 puis 6 - Nombre d'itérations = 2 - Algorithme de l'échantillonneur de Gibbs

Image 10 :

### Restauration des contours d'une image satellite



Figure 1 : Image de scène réelle

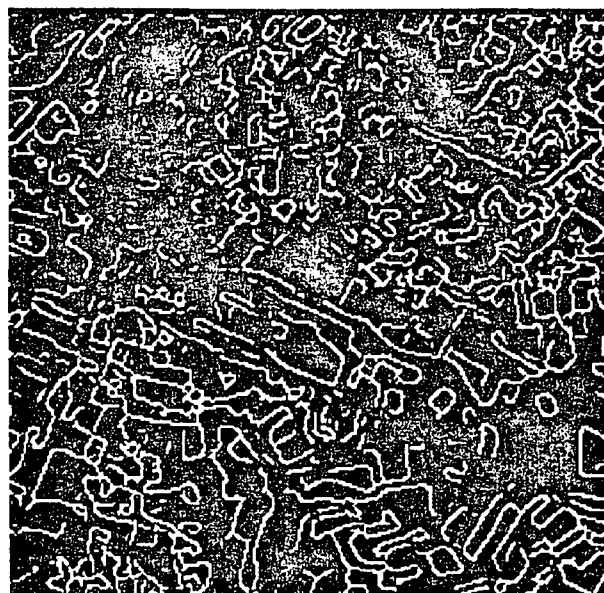


Figure 2 : Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima

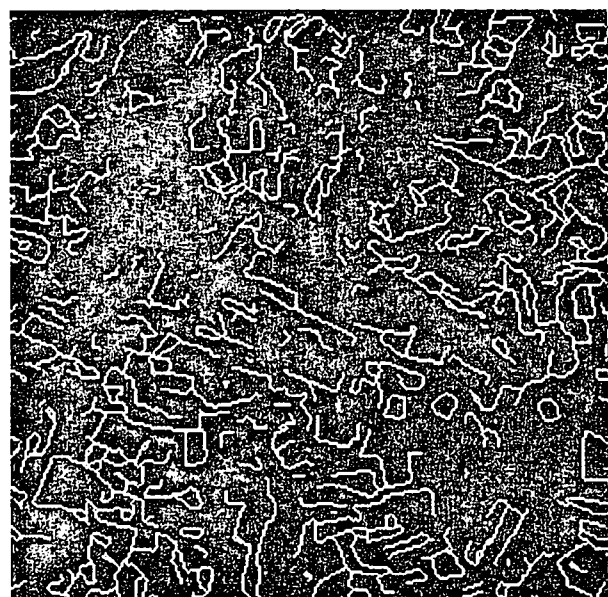


Figure 3 : Restauration des contours incomplets - Période fixe = 3 - Nombre d'itérations = 1 - Algorithme de l'échantillonneur de Gibbs

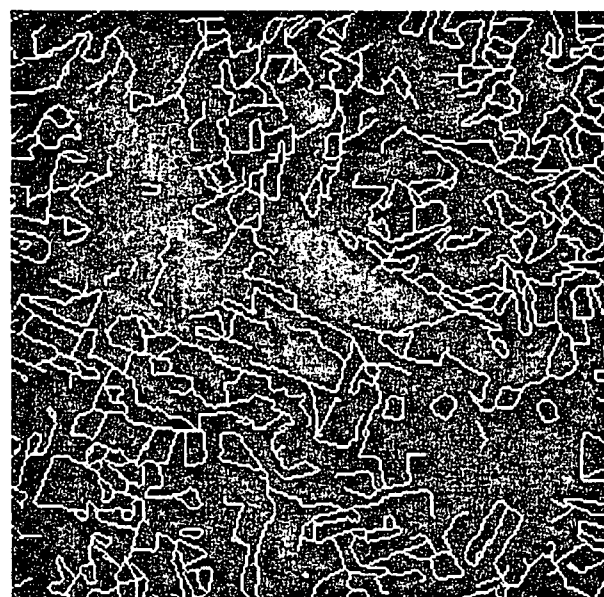


Figure 4 : Restauration des contours incomplets - Période variable = 2 puis 5 - Nombre d'itérations = 2 - Algorithme de l'échantillonneur de Gibbs



Image 11 :

### Restauration des contours d'une image satellite

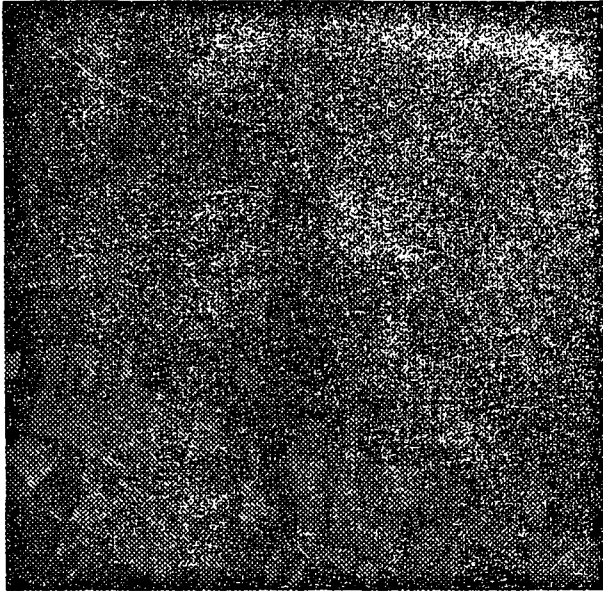


Figure 1 : Image de scène réelle



Figure 2 : Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima



Figure 3 : Restauration des contours incomplets - Période fixe = 2 - Nombre d'itérations = 1 - Algorithme de l'échantillonneur de Gibbs



Image 12 :

### Restauration des contours d'une image satellite

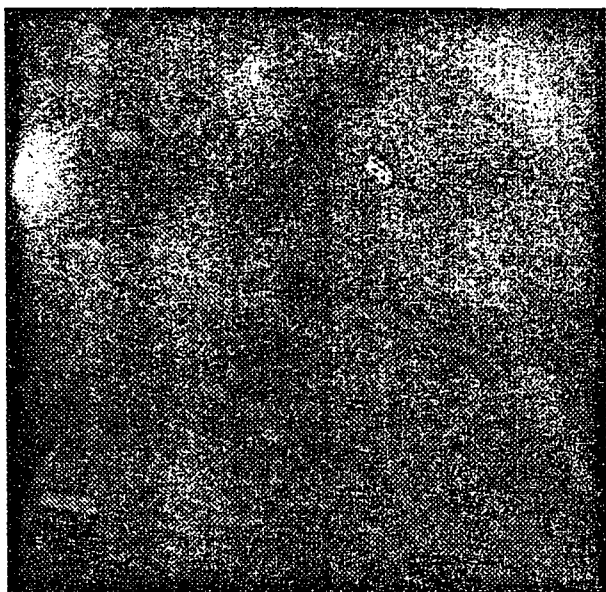


Figure 1 : Image de scène réelle



Figure 2 : Contours détectés après filtrage de Canny-Deriche et suppression des non-maxima

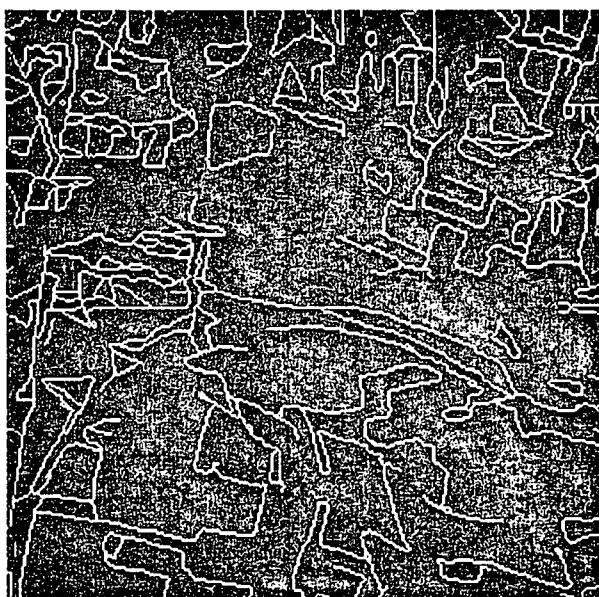


Figure 3 : Restauration des contours incomplets - Période variable = 3 puis 6 - Nombre d'itérations = 2 - Algorithme de l'échantillonneur de Gibbs

### 3.5.5 Temps d'exécution.

Pour chacune des images restaurées, nous indiquons :

- Le VPR : le nombre de pixels par processeur.
- Le temps d'exécution sur la CM[ 12].
- Le temps total.
- Le nombre d'itérations effectuées.

	VPR	Temps CM	Temps total	Nombre d'itérations
Image 1	2	620,84 sec	640,35 sec	2
Image 2	2	278,38 sec	314,01 sec	1
Image 3	2	1311,73 sec	1329,33 sec	2
Image 4	2	284,29 sec	299,92 sec	1
Image 5	2	965,59 sec	991,45 sec	1
Image 6	8	1194,81 sec	1226,24 sec	1
Image 7	8	2539,83 sec	2658,78 sec	1
Image 8	8	1696,88 sec	1779,01 sec	1
Image 9	8	2375056 sec	2376,32 sec	2
Image 10	8	2231,25 sec	2259,29 sec	2
Image 11	8	3777,66 sec	3812,9 sec	3
Image 12	8	2346,54 sec	2367,97 sec	2

Le nombre d'itérations à effectuer est souvent plus important lorsqu'on met en oeuvre la méthode de la période variable.

#### **4. Conclusion**

La méthode proposée permet la restauration d'image de contours incomplets et a pour intérêt de pouvoir compléter des images de contours présentant un manque important d'information.

Voici un bref résumé de la méthode utilisée :

Une classification des éléments de contours (états) et des différents types de connexions entre deux états est déterminée. Seules les interactions locales au niveau du pixel sont prises en compte, mais chaque site détient une information plus globale : la direction macroscopique.

Pour compléter les données manquantes, on effectue successivement des prolongation et des raccourcissement de lignes de contours, ceci dans le but d'obtenir une configuration stable où apparaissent des "T-jonctions".

Deux algorithmes basés sur une modélisation par champs de Markov de l'image ont été mis en oeuvre : un algorithme stochastique ( l'échantillonneur de Gibbs [4] ), et une méthode déterministe (ICM [1]). L'algorithme déterministe est plus rapide et donc mieux adapté, lorsque les données ne sont pas bruitées car elle ne contiennent pas de "fausse information".

Des améliorations restent à apporter. En effet, pour des images initiales bruitées, les résultats obtenus ne sont toujours satisfaisants.

## 5. Références.

- [1] J. Besag, "On the statistical analysis of dirty pictures", *JL of Royal Statistical Society*, series B, Vol. 68, pp 259-302, 1986.
- [2] A. Blake et A. Zisserman, "Visual reconstruction", *MIT Press, Cambridge - MA*, 1987.
- [3] D. Geiger et F. Girosi, "Parallel and deterministic algorithms for MRFs : surface reconstruction and integration", *Proc. ECCV*, Antibes, Avr. 1990.
- [4] G. Geman et D. Geman. "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images", *IEEE Trans. Pattern Analysis, Machine Intel.*, Vol. PAMI-6, p 721-741, Nov 1984.
- [5] J.L Marroquin, "A Markovian Random Field of Piecewise Straight Lines", rapport interne du "*Centro de Investigacion en Matematicas*", Mexico , 1989.
- [6] J.L Marroquin, "A Markovian Random Field of Piecewise Straight Lines", *Biological Cybernetics* , p 457-465, 1989.
- [7] P. Van Laarhoven et E. Aarts, "Simulated Annealing: Theory and Applications" *D. Reidel Publishing Company*, Dordrecht, Holland, 1987.
- [8] J. Zerubia et C. Graffigne, "Rapport sur la Segmentation : Approches Statistiques", *Rapport "Segmentation des images" du Gdr n° 134 TDSI*, Mai 1991.
- [9] J. Zerubia et F. Ployette, "Détection de contours et lissage d'image par deux algorithmes déterministes de relaxation. Mise en oeuvre sur la machine à connexions CM2", Article *Traitement du signal*, Vol.8 - n°3, p 165-179, 1991.
- [10] J. Zerubia, Cours du D.E.A Vision : "Restauration d'Image", 1990.
- [11] "The Connection Machine System", CM User's Guide, Version 6. ", *Thinking Machines Corporation*, Cambridge, Massachusetts, Nov. 1990 .
- [12] "C\* Programming Guide, Version 6. ", *Thinking Machines Corporation*, Cambridge, Massachusetts, Nov. 1990 .



**ISSN 0249 - 6399**